NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

ANALYSIS OF ERROR PROPAGATION WITHIN HIERARCHICAL AIR COMBAT MODELS

by

Salih Ilaslan

June 2016

Thesis Advisor: Thomas W. Lucas
Second Reader: Jeffrey Appleget

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2016	**3. REPORT TYPE AND DATES COVERED** Master's thesis	
4. TITLE AND SUBTITLE ANALYSIS OF ERROR PROPAGATION WITHIN HIERARCHICAL AIR COMBAT MODELS		**5. FUNDING NUMBERS**	
6. AUTHOR(S) Salih Ilaslan			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000		**8. PERFORMING ORGANIZATION REPORT NUMBER**	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A		**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited		**12b. DISTRIBUTION CODE**	

13. ABSTRACT (maximum 200 words)

Operations research analysts often use a hierarchy of combat models to provide insight to military decision makers. Briefly, lower-level, higher-resolution models provide input to higher-level, lower-resolution models. This allows analysts to explore how engineering and tactics changes can affect campaign effectiveness. This thesis builds upon previous research and examines various methods for employing distributions of engagement-level model outputs as input to campaign-level models, instead of just using the average. We contrast methods for "linking" the engagement-level model to the campaign-level model. Previous research indicates that when expected values alone are propagated through layers of combat models, the final results will likely be biased, and risk underestimated.

An air-to-air engagement model is developed to generate a data library that is used as input in a stochastic Lanchester campaign model. A variety of sampling methods are employed to sample from the engagement model's output data library to provide input to the campaign model. The results indicate that the manner in which the engagement and campaign models are linked has substantial impact on the estimates of operational effectiveness and risk. Additionally, our research illustrates how running a designed experiment on the engagement-level model, to generate a library of data that can be linked to the campaign-level model, can support robust decision making.

14. SUBJECT TERMS hierarchical combat modeling, air combat modeling, campaign analysis, mean and variance analysis, sampling methods, metamodeling, error propagation, Lanchester equations, agent-based simulation, design of experiments, simulation output analysis			**15. NUMBER OF PAGES** 133
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

ANALYSIS OF ERROR PROPAGATION WITHIN HIERARCHICAL AIR COMBAT MODELS

Salih Ilaslan
Captain, Turkish Air Forces
B.S., Turkish Air Force Academy,
2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
June 2016

Approved by: Thomas W. Lucas
Thesis Advisor

Jeffrey Appleget
Second Reader

Patricia A. Jacobs
Chair, Department of Operations Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Operations research analysts often use a hierarchy of combat models to provide insight to military decision makers. Briefly, lower-level, higher-resolution models provide input to higher-level, lower-resolution models. This allows analysts to explore how engineering and tactics changes can affect campaign effectiveness. This thesis builds upon previous research and examines various methods for employing distributions of engagement-level model outputs as input to campaign-level models, instead of just using the average. We contrast methods for "linking" the engagement-level model to the campaign-level model. Previous research indicates that when expected values alone are propagated through layers of combat models, the final results will likely be biased, and risk underestimated.

An air-to-air engagement model is developed to generate a data library that is used as input in a stochastic Lanchester campaign model. A variety of sampling methods are employed to sample from the engagement model's output data library to provide input to the campaign model. The results indicate that the manner in which the engagement and campaign models are linked has substantial impact on the estimates of operational effectiveness and risk. Additionally, our research illustrates how running a designed experiment on the engagement-level model, to generate a library of data that can be linked to the campaign-level model, can support robust decision making.

THIS PAGE INTENTIONALLY LEFT BLANK

THESIS DISCLAIMER

The reader is cautioned that the computer programs presented in this research may not have been exercised for all cases of interest. While every effort has been made, within the time available, to ensure that the programs are free of computational and logical errors, they cannot be considered validated. Any application of these programs without additional verification is at the risk of the user.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AIM	air intercept missile
AMRAAM	Advanced Medium Range Air-to-Air Missile
AWACS	Airborne Warning and Control System
BVR	beyond visual range
C2	command and control
CAP	combat air patrol
COSMOS	Complex Systems Modelling and Simulation
DOE	design of experiments
DP	design points
GCAM	General Campaign Analysis Model
HPC	High Performance Computing
JMEM	Joint Munitions Effectiveness Manual
LCOM	Logistics Composite Model Simulation
MANA	Map Aware Non-Uniform Automata
MCET	Mine Warfare Capabilities and Effectiveness Tool
MOE	measure of effectiveness
MOP	measure of performance
NOLH	nearly orthogonal Latin hypercubes
NSS	Naval Simulation System
OA	operations analysis
OPNAV	Office of the Chief of Naval Operations (US)
OR	operations research
R5FF	resolution five fractional factorials
SAM	surface to air missile
SEED	Simulation Experiments and Efficient Designs
STORM	Synthetic Theater Operations Research Model
TUR	Turkey

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Combat simulations are widely used to simulate tactical to campaign operations to provide insight to decision makers. These simulations vary in the amount of detail they contain, and are sometimes referred to as "lower level and higher resolution" or "higher level and less resolution." The higher the level, the less the detail, but the greater and more aggregated the span of simulated operations. All combat models require input data, sometimes estimated from the result of live tests or exercises, or, as we discuss, sometimes from the output of lower-level combat simulations. The process of utilizing lower-level model outputs as input to a higher-level model is referred to as a hierarchy of models. This hierarchy allows the analyst to explore how engineering and tactics changes can affect campaign effectiveness. Operations research analysts of many countries utilize this hierarchy of simulations to provide insight to their leadership. Instead of conducting real world exercises to decide which platforms to procure, hierarchical simulation is a useful tool to explore engineering effects on missions and campaigns since there is no potential for loss of money and lives. By using hierarchal simulation, the Turkish Air Force can gain insight into questions such as "how will better sensors or weapons affect the outcome of a series of air battles?" Senior leadership can then use this information to determine the best investments to achieve and sustain warfare dominance within a particular budget.

This thesis builds upon previous error propagation research conducted by U.S. Navy LT Russell Pav (2015) and examines various methods for employing distributions of engagement model outputs as inputs to campaign models, instead of just using the average. We contrast methods for "linking" the engagement-level model to the campaign model. Previous research indicates that when expected values alone are propagated through layers of combat models, the final results will likely be biased, and risk underestimated.

For this thesis, we first develop an engagement-level model for a two-versus-two air engagement between jet fighters in the stochastic, agent-based Map Aware Non-uniform Automata (MANA) simulation environment (McIntosh, 2009). The measures of

performance (MOPs) for this model are based on open source operating characteristics of jet fighters—such as speed, range, and weapon probabilities of kill. These are inputs to the MANA simulation. The measures of effectiveness (MOEs) are losses, time of battle, and probability each side wins; these are the outputs from the engagement-level model. Variability in the MOPs is induced and explored using two different designs of experiments (DOEs), the Nearly Orthogonal Latin Hypercube (NOLH) and the Resolution V Fractional Factorial (R5FF), and we examine how the choice of experimental design impacts results. The MANA runs yield a library of outputs that can be accessed as needed by the campaign model.

Next, we construct a stochastic Lanchester Linear Law campaign model. Each campaign simulates four discrete engagements of 25 versus 25 jet fighters, and each engagement utilizes a breakpoint such that the engagement is terminated as soon as one side is depleted to a quarter of their original strength. The attrition coefficients for the stochastic Lanchester campaign model are determined by the losses and time of battle obtained from the engagement-level MANA model. Because MANA is stochastic, many replications are run, producing a distribution of outcomes. However, the campaign-level model takes scalars as input. Thus, the research question is how best can we link these two models to account for the variability? Several methods for using the engagement-level results as inputs to the campaign-level simulation are assessed and compared in this thesis.

This research aims to answer following research questions:

- What are the results of using different methods (e.g., taking the mean, random sampling, metamodeling, etc.) to convert outputs from a lower-level air combat model to inputs to a campaign-level model?
- Is using a lower-level model's output as input to a higher-level model a reasonable approach for air combat models?
- Does use of the design generated from NOLH and R5FF methods cause a significant change in outputs?

Analysis indicates that the sampling methodology and the manner in which the engagement and campaign models are linked have a significant impact on the estimate of operational effectiveness and risk. Beyond conducting random sampling on a different

scenario than the one Pav used, this study employed additional techniques to sample from or link to the lower-level engagement model. In particular, we explored and compared the embedded metamodeling and linked metamodeling approaches, and established a key difference between the two; namely that the deterministic embedded metamodeling technique can lead to biased results and underestimation of risk. For example, Figure 1 illustrates the difference between the deterministic embedded metamodeling approach and the linked metamodel approach, for a simple one-variable experiment. The blue dots connected by the smoother line represent the result of evaluating a deterministic composite formula for P(Win) that is a function of system-level variables at multiple design points. The formulas were developed by finding a regression response at each modeling level. Since this example only varied one of the factors (variables), the other variables were fixed at their baseline levels. The red dots with the standard deviation error bars represent variability in the estimate for P(Win) with the models directly linked. That is, a sample of engagement-level outputs, across many design points, was run in multiple replications of the stochastic Lanchester campaign model. The results show that the deterministic embedded metamodel would usually result in overestimating P(Win) and underestimating risk.

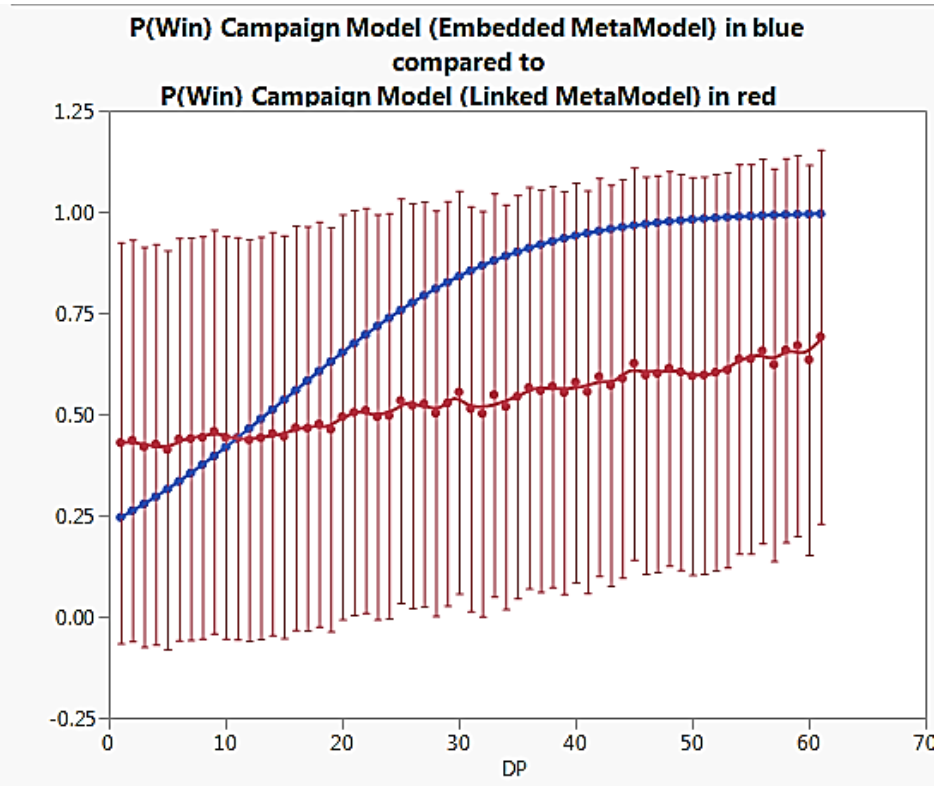


Figure 1. Comparison of Deterministic Embedded Metamodel results (blue) to Linked Metamodel results (red)

Lastly, we emphasize that an excellent method for systematically exploring uncertainties in the scenario is design of experiments; and the resulting output is capable of adequately characterizing variability and risk. In our experimentation, we used both the Nearly Orthogonal Latin Hypercube (NOLH) and the Resolution 5 Fractional Factorial (R5FF) designs. If time permits, it would be desirable to supplement the desirable features of the space-filling NOLH with the corner-sampling R5FF. In Chapter IV, we demonstrated how the campaign model linked to DOE output is much more effective and useful for quantifying risk than the use of the deterministic embedded metamodel approach. This is because the designed experiment captures the variability that results when engineering factors, many of which we may be uncertain about or do not have control over, are varied over reasonable ranges.

References

Pav, R. (2015). *Experiments in Error Propagation Within Hierarchical Combat Models.* Master's thesis, Naval Postgraduate School, Monterey, CA.

McIntosh, G. C. (2009). *MANA-V (Map aware non-uniform automata – Vector) supplementary manual*. Auckland, New Zealand: New Zealand Defence Technology Agency.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

It was a long, laborious but enjoyable journey in Naval Postgraduate School's Operations Research Program. First of all, I would like to express my deepest appreciation to the citizens of Turkey, whose taxes were spent for my education here. Next, I want to thank my thesis advisor Professor Thomas W. Lucas and my second reader Dr. Jeffrey Appleget for their mentorship and guidance. It was an honor for me to work with them. In addition, I would like to thank Research Associate Mary McDonald for imparting her knowledge and expertise in this study. Through their patience and gentle hearts, they gave me the power to do my research and motivated me to complete this thesis.

During my life at NPS, I met great people and walked this way with their companionship. In bad and good, I was lucky to have them in my life. I would like to express my special gratitude and thanks to Ersin Ozmen, Halis Can Polat, Jennifer Kuramura, Muhammet Tekin, Tacettin Ersoz, Samet Salin, Serif Kaya, Sophia Bay, Peter Bazalaki, Emilie Monaghan, John Olabode, Asim Aslan, Serkan Aktas and many others who I couldn't mention here for their endless support and kind, understanding spirit. Memories will never be forgotten.

Additionally, I would particularly like to thank my mother for her endless support and love.

Lastly, I would like to thank the staff members of Naval Postgraduate School Department of Operations Research for their expertise and patience. I received great academic instruction from the professors at this institution and am returning to Turkey with rewarding knowledge.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

"Peace at home, peace in the world."

—Mustafa Kemal Ataturk

Models are an essential element in planning and resourcing for air combat. A model, whether in the form of a diagram, a set of differential equations, or a computer simulation, permits us to generate information as to possible outcomes before we engage resources. Because we have come to depend on models to provide insight and generate solutions, the data used in computer simulated air combat modeling must be as precise as possible. As a method used in air combat modeling, hierarchal combat modeling is widely used to estimate the effects of engineering changes on the outcomes of a campaign. To do that, high-resolution low-level models provide inputs to low-resolution higher-level models. During this process, stochastic low-level models produce outputs with probability distributions, and, if the variance of this distribution is not taken into account, when the data is applied to higher-level models, small errors can become much larger errors. This phenomenon is known as error propagation within hierarchal combat models. Error propagation can cause uncertainty regarding the outcomes of a campaign, which is a risk that decision-makers should assess carefully (Pav, 2015).

In this chapter, we first define combat modeling and give reasons for the use of combat modeling. In the literature review section, we discuss air combat modeling as it pertains to this study. Then, we review hierarchal modeling applications in both combat modeling and other scientific areas. We also discuss different model levels associated with hierarchal modeling and error propagation/assessment. Finally, research questions and the scope of the thesis are presented, including our objectives and methodology.

A. BACKGROUND: DEFINITION AND NEED FOR COMBAT MODELING

Today's military operations are complex and often risk both money and human lives. Thus, understanding combat is critical. Towards that end, analysts around the world

use combat modeling as a tool to discover solutions for real life problems. In their book, *Combat Modeling*, Alan Washburn and Moshe Kress define a model as "A model is an abstraction of reality" (Washburn & Kress, 2009, p. 1). They go on to explain that "Our limited intellects permit us to deal only with abstractions that retain the essence of the matter without the distracting details. As a great tool, models are used for reasoning, insight, planning, and prediction. They need to capture the key factors of the object or situation and faithfully represent them so that the models can be utilized effectively" (Washburn & Kress, 2009, p. 1). Having the goal of solving real problems, these models can be a set of differential equations or a diagram or a computer simulation model. In order to effectively use this tool, operations research analysts strive to solve real life problems by generating models that include relevant details of these problems. The utility of these models is that they permit us to investigate situations without the risk of losing money and lives.

An important early combat model was developed by Frederick W. Lanchester during his studies on attrition during World War I. Lanchester, an engineer and a car company owner in London, was interested in aerial battles and developed differential equations to predict the outcome of battles (Lanchester, 1916). Since then, many combat models have been generated and used for similar purposes. In practice, most combat models are a description or representation of weapon systems and combat operations implemented in a computer program. Combat models are a crucial tool in the analysis of military operations, tactics, and strategies.

Testing a weapon system by employing combat modeling simulations is significantly less expensive than experiments using real weapon systems: "Combat models provide information that assists decision-makers in making and justifying decisions that involve the expenditure of billions of dollars and impact many lives" (Thomas W. Lucas, Turker Turkes, 2004, p. 1). For instance, the Turkish Air Force's capabilities are mostly based on Lockheed Martin's F-16C and F-16D multirole fighter aircraft (IHS Jane's, 2016). The unit cost of a single F-16C aircraft is $18.8 million (fiscal 1998 constant dollars) (U.S. AIR FORCE, 2015). Moreover, Turkish F-16Cs carry AIM-9S/X Sidewinder (Air Intercept Missile) and AIM-120C AMRAAM (Advanced

Medium Range Air-to-Air Missile) missiles for air-to-air operations. The cost of a single AIM-120C AMRAAM missile is about $1.2 million and the cost of a single AIM-9S/X Sidewinder missile is about $600,000 (IHS Jane's, 2016). In order to evaluate the tactical effectiveness of these missile systems in combat, performing a real exercise would be very costly and limited in scale. Instead of live exercises, using high-resolution combat models provides helpful insights about the tactical effectiveness of these aircraft and missile systems. Moreover, combat modeling can be used when the systems do not actually exist. That is, they can be used to help guide investment decisions in developing future systems.

In addition, combat modeling does not require actual military assets and personal. Hence, there is no loss of military assets or casualties when running a computer-based combat simulation. Since all scenarios are executed on the computer, there is no need to have an actual enemy force to fight against

While combat models are useful in providing insight about military issues or weapon systems, they do not provide precise and actual outcomes of a real campaign. Therefore, analysts should be careful when building and using combat models, determining the inputs, and analyzing the outputs of the model. These fundamental steps of modeling must be studied carefully; otherwise, outcomes of a combat model may lead decision-makers to wrong choices—which may cause the loss of many lives.

As one of the methods of combat modeling, hierarchal combat modeling is used by today's military forces in many countries to investigate the effects of engineering level changes on the outcomes of a campaign, to analyze alternative force structures, and to learn about which weapons and weapon platforms are crucial to develop and procure. In the following literature review section, previous studies and more detailed information about hierarchal combat modeling is presented.

B. LITERATURE REVIEW

This section contains a literature review and explains air combat modeling and hierarchal combat modeling in the context of this research.

1. Air Combat Modeling

As a branch of today's warfare, air combat contains many interacting systems that are operated by both human and command/control (C2) operators. To assess the effectiveness of aerial systems, "Computer-based air combat modeling is a powerful tool, widely accepted for its usefulness. The extremely high cost of operating aircraft and their weapons has led to a rapid growth in the development and use of computer simulation models as a basis for tactics development, pilot training, and operational evaluation of weapon systems" (Rao, Lucas, Morley, Selvestrel, & Murray, 1993). Additionally, procuring, operating, and maintaining multi-integrated air combat systems are very expensive compared to the cost of many army vehicles and weapon platforms. For instance, a German leopard tank's cost is $6,790,000, whereas a F-35A multi-role fighter aircraft is $90,000,000 (IHS Jane's, 2016). To reduce cost, simulation modeling provides an opportunity for investigating the effects of technological progress on the effectiveness of current and potential aerial systems. Briefly, air combat modeling is a specified field of combat modeling where air-to-air or air-to-ground operations are modeled.

Reaping the benefits of air combat modeling often requires time-consuming and complex analysis since these models require specific inputs matching the capabilities and tactics in air operations. Interactions between new aerial systems and current tactics must be analyzed carefully before decision makers determine whether or not to procure a new aerial platform or weapon system. Consequently, employing air combat modeling saves a lot of money; however, a substantial investment in time and effort is necessary for credible analysis.

From the beginnings of combat modeling as a campaign analysis tool, there have been many research studies that modeled air combat. As the author mentioned before, Frederick W. Lanchester's effort is one of the oldest air combat modeling studies. Lanchester published a book entitled *Aircraft in Warfare: the Dawn of the Fourth Arm* in 1916 and shared his ideas about aerial warfare (Lanchester, 1916). In the book, Lanchester described a family of differential equations that modeled how two military forces would attrite each other in combat. Lanchester developed two different models; he

believed that “square law,” one of the models, would describe “modern” air combat. These equations are described in the second chapter of this study.

There have been many studies on the validity of various forms of Lanchester’s laws over the years; for example, see Engel (1954), Hartley (2001), and Lucas and Dinges (2004). As a whole, these researchers find that no consistent Lanchester law dominates, but a combination of Lanchester linear and log laws fit best. Recently, three academicians at the University of York, Ian Horwood (Historian), Niall MacKay (Mathematical Physicist), and Christopher Price (Historian) analyzed historical data of various air campaigns (Battle of Britain, 1940; Pacific Air War, 1941–1945; Korean War, 1950–1953) and found results opposed to what Lanchester believed. Their starting hypothesis was that air combat is a set of duels, best modeled as random with the Lanchester linear law. They postulated that air combat does not obey a square law, based on their analysis of historical air combat data. They concluded that air combat is 80% linear-law and 20% asymmetric (Horwood, Mackay, & Price, 2014). As a result of their conclusion, this study employs a linear law in the campaign-level model instead of the more traditional square law.

After Lanchester’s studies, many air combat modeling equations, tools, and simulation programs have been developed based on derivatives and extensions of Lanchester’s original two laws. The U.S. Air Force is currently using many different simulation programs in campaign analysis, depending on the size of the campaign or the operation that they are interested in and study objectives. BRAWLER is one of these simulation programs. In the website of Defense Systems Information Analysis Center of U.S. DOD (https://www.dsiac.org), BRAWLER is described:

> … simulates air-to-air combat between multiple flights of aircraft in both the visual and beyond-visual-range (BVR) arenas. … This simulation of flight-vs.-flight air combat is considered to render realistic behaviors for military trained fighter pilots. BRAWLER incorporates value-driven and information-oriented principles in its structure to provide a Monte Carlo, event-driven simulation of air combat between multiple flights of aircraft with real-world stochastic features. The user decides the pilot’s decision process, including doctrine, mission-specific objectives, and tactics; level or training and experience; and perceived capability of the enemy. (Brawler, 2015)

In Figure 1, a visualization of an air-to-air combat model in BRAWLER is shown.

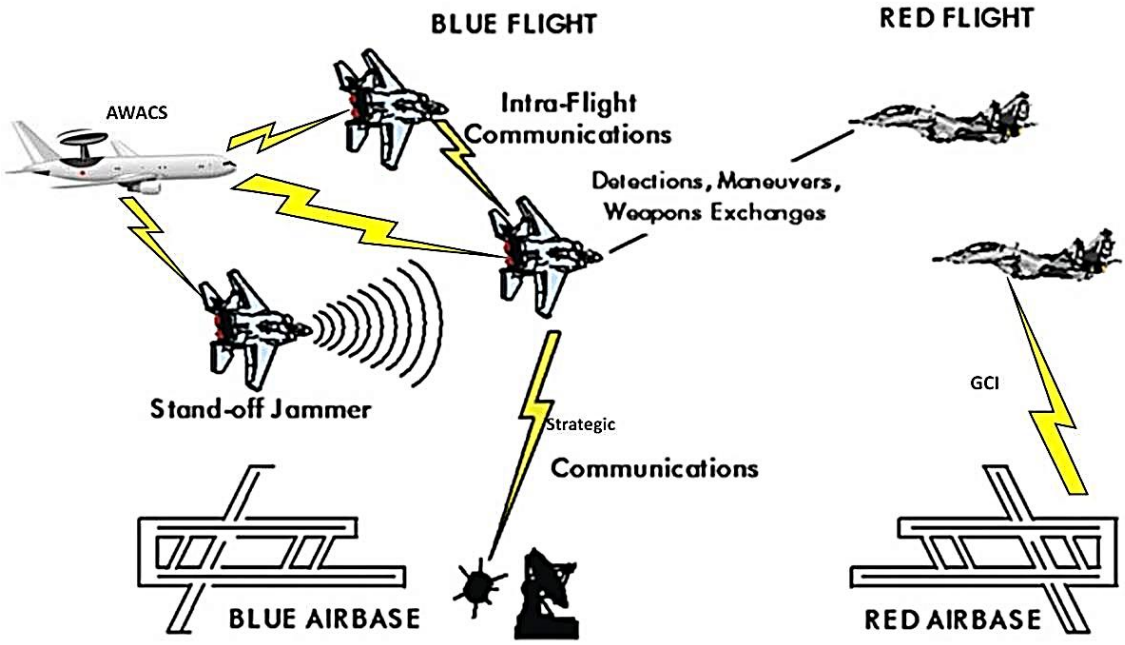


Figure 1. Air-to-Air Combat Model in BRAWLER. Source: Brawler (2015)

Another simulation program that is currently used by the U.S. Navy, U.S. Air Force, and Turkish Air Force is the Synthetic Theater Operations Research Model (STORM). STORM "is the primary campaign analysis tool used by the Office of the Chief of Naval Operations, Assessment Division (OPNAV N81) and other Department of Defense organizations to aid in providing analysis to top-level officials on force structures, operational concepts, and military capabilities" (Seymour, 2014, p. v).

In many OPNAV N81 studies, the stochastic, mission-level model BRAWLER is run to generate input for the stochastic, campaign-level simulation STORM. In this thesis, we employ a simpler pair of engagement-level and campaign-level models of air-to-air combat to investigate error propagation within hierarchal air combat models.

2. Hierarchical Modeling Approach

As a type of simulation modeling, hierarchal modeling has been used in many scientific studies in different science fields, such as environmental science, social science, and military science, etc. For instance, hierarchal modeling has been widely used in epidemic disease analysis, where the model can map the possible direction and the number of people potentially affected by the disease (A. B. Lawson, 2013). The methods and analysis utilized in these studies can be different, but the basic idea is similar. Lower-level models provide inputs for upper-level or aggregate models. Christopher K. Wikle described the need for hierarchal modeling in environmental science in one of his studies: *Hierarchal Models in Environmental Science*. He stated that environmental systems include spatial-temporal processes and they are interacting with different scales since they are very complicated. The processes such as monitoring networks, computer models, remote sensing platforms, and geographical information systems generate a large amount of data (Wikle, 2003). Wikle mentioned that it is not enough to evaluate such processes with a joint perspective. Therefore, there is an indispensable need of hierarchal modeling to provide a coherently connected system of conditional circumstantial models and return reasonable outcomes for given processes.

Another important usage of hierarchal modeling has been with military simulations. In the military simulation world, operations research analysts of many countries explore the effects of engineering-level changes on campaign output by employing hierarchal combat modeling. Involving the detailed and sophisticated features of weapon platforms, high-resolution lower-level models, such as those of one-on-one combat, are readily simulated with current simulation software programs. Then, campaign-level models use the output of these higher-resolution models as input. Simulating campaign-level combat is a very complex and time-consuming study, and must be done at a less detailed (i.e., more aggregated) level of resolution. It is important to address both the detailed aspects of combat represented in high-resolution models (such as engagement-level and mission-level models) and the broader context of campaigns.

In order to conduct a useful and reasonable campaign analysis, objectives, tasks, and metrics of the campaign must be well defined. Essential details of the mission should be taken into consideration, analyzed, and included as an assumption in the campaign analysis process. Therefore, many countries are using hierarchal families of combat models in their studies. Typically, they employ campaign-level models that are fed by the outputs of mission-level models, which in turn are fed by the inputs of even more detailed models. This hierarchy among simulations is often displayed as a pyramid. The foundation of the pyramid consists of many highly-detailed and narrowly focused physics-based models. The top of the pyramid typically consists of a single highly-aggregated campaign-level model. A version of this pyramid that OPNAV N81 uses is shown in Figure 2.

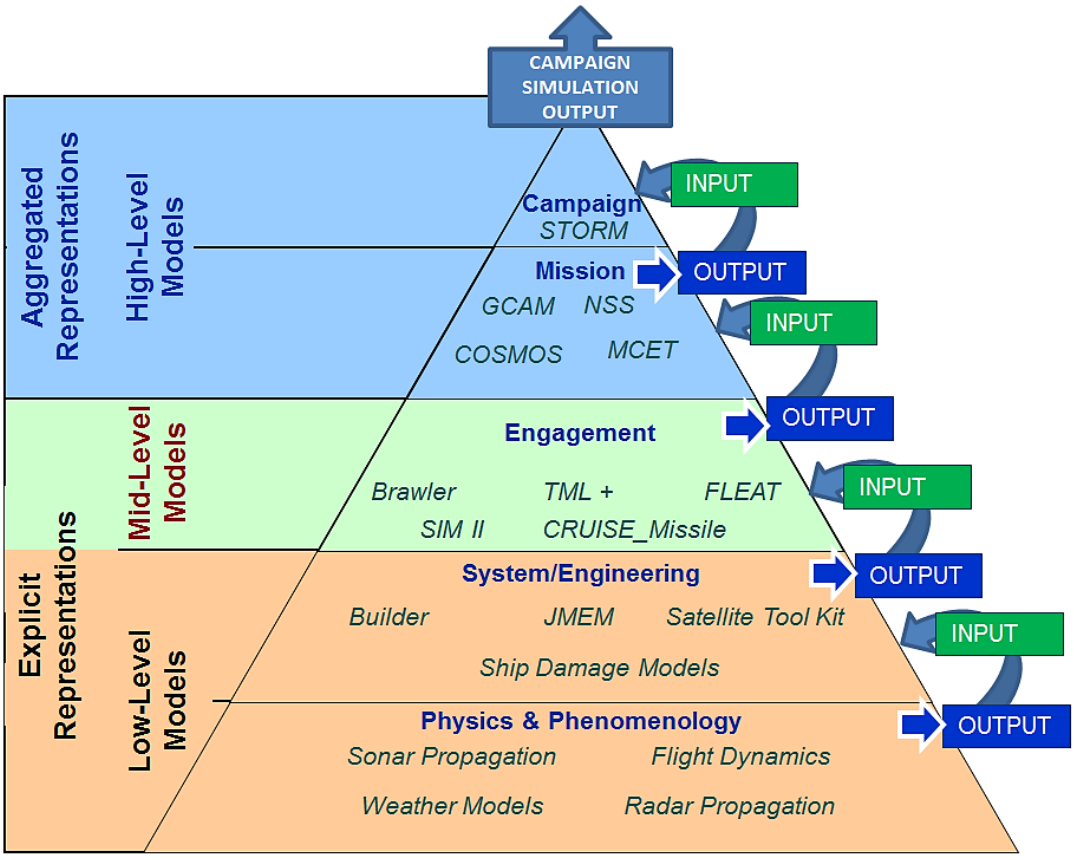


Figure 2. Hierarchy Pyramid of Simulation Models. A Similar Version is Used by OPNAV N81

As an example of hierarchal combat modeling, one of outputs of one-on-one aircraft engagements, such as a missile's probability of kill, are generally used as input for a mission-level model. The output of this mission-level model (say an exchange ratio) may then be used as an input to a campaign-level simulation.

By utilizing hierarchal combat modeling, decision makers can observe the effects of changes in weapon platforms on tactical performance and campaign effectiveness. As well as many other countries, the U.S. Air Force and U.S. Navy are using hierarchal combat modeling to determine the force structure required to meet future military needs. The U.S. Air Force Analysis Panel's presentation slides on an analytic framework and analytic agenda show (see Figure 3) a similar hierarchal pyramid of simulation models (Cerniglia-Mosher, 2009).

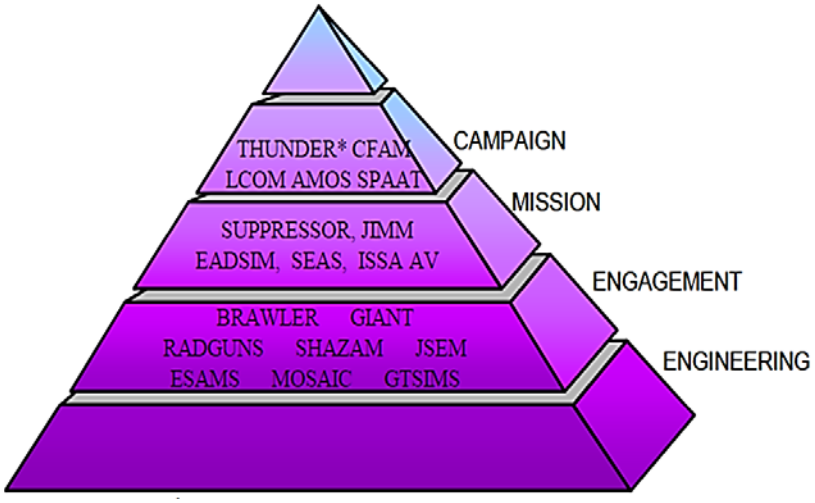


Figure 3. Hierarchical Combat Modeling Pyramid. Source: Cerniglia-Mosher (2009)

All of this begs the question: why not put all of the detail into one campaign-level model? Simply put, it is infeasible given time and processing constraints to run a high-resolution campaign-level simulation over the necessary replications to generate the required output. Moreover, the complexity of doing so is staggering (Lucas and McGunnigle 2003). At this point, hierarchal combat modeling allows analysts to link low-level details to campaign-level outcomes—and saves a lot of time and money, in

comparison to conducting live exercises. This process enables the objectives, tasks, and metrics of the campaign to be analyzed and included in each model level. The fundamental levels of hierarchal modeling are discussed in the following paragraphs of this thesis.

a. Engagement /Engineering Level Models

Operations research analysts study the quantitative characteristics of military assets, such as probability of hit, radar range, missile range, etc. In order to determine how effectively military assets perform in tasks and missions, these characteristic quantities are captured using the output of highly-detailed engineering-level simulations (T.W. Lucas, personal communication, September 9, 2015). These kinds of models are known as entity, engineering, or engagement-level models.

Additionally, these models are useful since the physical testing of military equipment is very costly. As it was mentioned before, conducting a military exercise with a loaded aircraft can cost millions of dollars. These high costs prevent armed forces from conducting more than a handful of live fire exercises and tests to evaluate their effectiveness. Before proceeding to the next step (i.e., building prototypes of military assets, missiles, and conducting live tests) engineering-level models may provide information to ensure that the design is reliable and plausible. Such an approach can save millions of dollars.

b. Mission-Level Models

Any kind of military operation involves small- or large-scale missions. For each mission type, several tactics can be developed and used in these operations. Before applying these tactics in related operations, they should be examined to see how effective and plausible they are. Therefore, live-fire exercises are an option for armed forces to test their tactics. However, conducting live-fire exercises to test these tactics is very expensive and may end up with the loss of costly equipment and lives. As a result, mission-level combat simulations are often used to evaluate the effectiveness of specific military assets and tactics in achieving mission objectives. The results of mission-level simulations are used by armed forces to develop tactical doctrine and to assist military

personnel in learning how to use military platforms, such as aircraft, helicopters, and missiles. Lucas stated in his lecture notes that mission-level models are commonly used to study doctrine, plan missions, assess force employment options, and evaluate force modernization choices (T.W. Lucas, personal communication, September 9, 2015).

c. Campaign-Level Models

Campaign-level models are useful not only in simulating campaigns and operations, but also in shaping future force structures. In the military world, experience has shown that the acquisition of a major weapon platform regularly takes more than decade. For example, Turkey signed a contract to procure 100 F-35A single-engine, single-seat, stealth multirole fighters for $175M each on 11 June 2002 (United States Government Accountability Office, 2004). After many years, Turkey was able to order six of these fighters, which are assured to be delivered by 2018. The rest of the aircraft are scheduled to be delivered at a rate of 10 fighters per year in subsequent years. This process showed that it will take almost 15 years to get the first F-35A aircraft and will take 10 years more to receive all 100 F-35A fighters. The Turkish Air Force plans to operate these multirole aircraft for half a century. Thus, these aircraft are a major pillar in Turkey's security. There will likely be future unanticipated conflicts which Turkey must be ready for over that time period. At this point, campaign models provide the ability to analyze the results of possible future conflicts given a specific force structure. Therefore, campaign modeling is a uniquely powerful tool in identifying capability gaps and in helping generate new and effective force structures based on future platform acquisitions. On the other hand, campaign models also allow armed forces to analyze the capabilities of weapons platforms currently in the force.

3. Error Assessment of Hierarchical Combat Modeling

By utilizing hierarchal combat modeling, decision makers can observe the possible effects of changes in weapon platforms on tactical performance and campaign effectiveness. Hierarchal combat modeling has proven beneficial to decision makers; however, this process may involve some error since the sample mean (average) outputs of lower-level models are typically used as inputs to higher-level models. Because the

outputs of the lower-level models generally have probability distributions, if the variability in lower-level models is ignored, then the outcomes of higher level model simulations may be biased or have understated variance (Lucas, 2000; Pav, 2015). Ultimately, the errors caused by using the means of lower-level models may result in biased and suboptimal decisions. As a consequence, operations research analysts cannot accurately quantify and assesses the risk. Sam Savage emphasized the importance of assessing risk in his research paper: *The Flaw of Averages Why We Underestimate Risk in the Face of Uncertainty*. Savage (2002) stated that "we need to stop thinking of uncertainties as single numbers—the average—and instead begin thinking of them as shapes, or distributions. And to deal with those distributions, we need to take advantage of modern computers for probability management" (p. 3).

C. RESEARCH PROBLEM AND RESEARCH QUESTIONS

As mentioned in the literature review section, using output from a lower-level model as input to a higher-level model by utilizing one of many possible sampling methods involves uncertainty. For instance, calculating a factor for a campaign metamodel by taking a sample mean of the output of an entity-level model is basically using point estimation, and therefore propagating forward merely a single point estimate. Since the sample mean is typically used, the variance of entity-level model output is not accounted for in the campaign metamodel. In other words, the uncertainty in the lower-level model is not propagated through higher-level models. This uncertainty may cause a bias in the outputs of the campaign model. Utilizing different sampling methods may decrease the uncertainty; however, no method has been shown to fully propagate variances through the hierarchal models in use by militaries around the world. Consequently, in order to provide insights about error propagation and generate alternative ways to reduce uncertainty, this research is guided by the following questions:

1. What are the results of using different methods (e.g., taking the sample mean, random sampling, design of experiment, etc.) to transform outputs from a lower-level air combat model into inputs to a campaign-level model?

2. Is using a lower-level model's output as input to a higher-level model a reasonable approach for air combat models?

3. Does use of the design generated from a space-filling nearly orthogonal Latin hypercube (NOLH) or a resolution V fractional factorial (R5FF) impact the results?

D. SCOPE OF THE THESIS

This study explores how the variance propagates through hierarchal air combat models using an engagement-level model and a campaign-level model. Several sampling methods are used to quantify the error propagation through our pair of hierarchal models. A two-on-two fighter aircraft engagement is selected as an engagement-level model and a basic engagement scenario is modeled using Map Aware Non-uniform Automata (MANA), an agent-based simulation (McIntosh, 2009). The outputs of the engagement-level model are fed into a stochastic Lanchester linear law campaign model by employing different sampling methods. The Python (an open source available at http://www.python.org.) programming language is used to simulate the stochastic Lanchester campaign. The outputs of the campaign Lanchester model include the probability of winning and the expected number of casualties when the campaign is won. These outputs are analyzed and compared using statistical tools such as Python and JMP(an open source available at http://www.jmp.com). Using a data library generated by executing designs of experiment on the MANA engagement model, regression analysis and other techniques are utilized to determine the effects of engineering-level changes on campaign output. Finally, the influence of the different sampling methods on risk assessment is evaluated, and suggestions are presented to enable adequate risk assessment.

THIS PAGE INTENTIONALLY LEFT BLANK

II. MODELING, SAMPLING AND ANALYSIS TOOLS

In this chapter, we present an overview of the modeling, sampling, and analysis tools used in this study. As previously mentioned, we use MANA to construct the engagement-level model scenario. To explore the effects of different combinations of performance characteristics on the aircraft's performance, both Nearly Orthogonal Latin Hypercube (NOLH) and Resolution V Fractional Factorial (R5FF) experimental designs are utilized. Both of these methods significantly downsize the required number of runs, as compared to a full factorial design that would test all possible combinations of the experimental factors (variables).

In this chapter, we briefly discuss the use of MANA, design of experiments (DOE), random sampling, and statistical metamodeling. We also describe the two software programs that we use for campaign-level modeling and statistical analysis.

A. USING NOLH AND R5FF EXPERIMENTAL DESIGNS TO EXPLORE FACTOR COMBINATIONS OF THE MANA ENGAGEMENT MODEL

Despite the fact that combat simulation modeling provides benefits and convenience for operations research analysts, it comes with substantial challenges. As mentioned in the literature review section, compared to live fire tests using military assets, these simulations are less expensive and dangerous. However, they are very complex, time-consuming, and require a deep knowledge of different computation programming languages and software. Due to the complexity of the phenomena being modeled, combat models tend to be exceedingly complex, often with thousands of input variables, many of which are uncertain. Moreover, they often take a long time to run even one replication. A single large-scale STORM run may take several hours to run. The large number of inputs of interest usually makes it infeasible to run all combinations of input variables that an analyst desires. To fully explore and evaluate an experiment with k variables, each with m levels, it would require a number of experiments equal to m^k multiplied by the number of stochastic replications per combination of variables (Sanchez, Lucas, Sanchez, Nannini, & Wan, 2012). This type of design, that explores all

possible combinations of variables in a brute-force manner, is called a full factorial. To give an example, suppose a simulation has 20 factors that an analyst wished to explore, each with three levels (say that corresponded to "low," "medium," and "high). In order to explore all of the factor combinations, 3^{20} runs, which is 3,486,784,401 (about $10^{9.5}$), is required. If each run takes just one minute, then running all combinations on a single processor would take 6,634 years. If replication is necessary, because the simulation is stochastic, then the time required must be multiplied by the number of replications needed. Luckily though, "Efficient design of experiments can break this curse of dimensionality at a tiny fraction of the cost," (Sanchez, Lucas, Sanchez, Nannini, & Wan, 2012). In order to break the curse of dimensionality, there are many options, and we explore the use of the Resolution V Fractional Factorial (R5FF) and the Nearly Orthogonal Latin Hypercube (NOLH). By carefully selecting input combinations, these designs reduce the number of simulation runs required to extract valuable information. By confounding estimates of higher order terms that are usually not of practical size or interest to analysts, R5FFs dramatically reduce the number of runs needed while providing the ability to estimate all main effects of the factors and all of their two-way interactions (Kleijnen, Sanchez, Lucas, & Cioppa, 2005). However, R5FFs only sample each factor (variable) at two levels, at the so-called "corners" of the input space. In order to reduce the number of required runs while simultaneously providing the analyst with analysis flexibility not provided by the use of the R5FF, NOLHs are often used (Cioppa and Lucas, 2007). NOLHs are efficient and space-filling designs that provide information throughout the experimental region, including the interior. They allow users to fit a variety of metamodels, including regression models with second and higher-order terms; most commonly, quadratic and two-way interaction effects. In Figure 4, we compare the space-filling capabilities of both design types for four factors with two levels each.

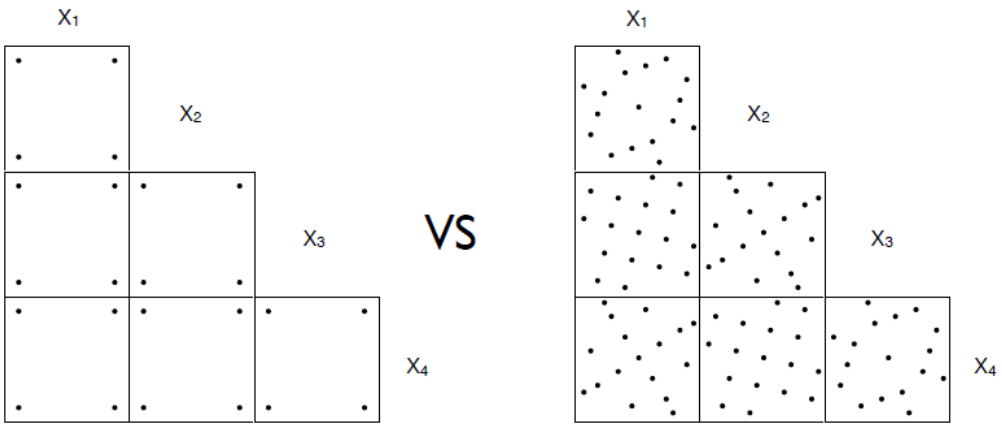


Figure 4. R5FF (16 DPs) vs. NOLH (17 DPs)

For this study, both the R5FF and the NOLH and are chosen to generate experimental designs for the MANA model. There are several reasons which led the author to use NOLH and R5FF designs. First of all, both designs are efficient in terms of required runs. However, the space-filling feature of the NOLH gives it a potential advantage in comparison to the R5FF design. In a particular, because it is space-filling, output from the NOLH can be used to identify nonlinear relationships and "knee in the curve" values as well as breakpoints and step functions. The R5FF is sometimes desirable because it tests at the more extreme "corner" points.

Previously, both of these experimental design methods were used in a thesis study by LT Russell G. Pav in 2015. In his thesis, he conducted an analysis to determine if the means and variances of MOEs of interest were statistically different by experimental design (Pav, 2015). To do that, he ran R5FF and NOLH experimental designs of experiment on his MANA simulation. Pav observed that the differences among the mean and standard deviations of MOEs were not practically significant. However, the terms in metamodels he built were not identical. In this study, we will examine if his findings remain true using a different scenario.

In summary, these designs allow the analyst to simultaneously fit complicated metamodels to the influential factors while exploring many variables for significance (Sanchez, S. M; Lucas; Sanchez, P. J.; Nannini; & Wan, 2012). As a result of this process, intelligent experimental design and the use of statistical metamodeling not only

reduce the simulation runs required to obtain useful output, but also enable us to analyze broad ranges of many factors.

B. ENGAGEMENT LEVEL MODELING TOOL: MANA

MANA (Map Aware Non-Uniform Automata) is an agent-based modeling environment that was developed at New Zealand Defense Technology Agency (DTA). MANA has been used in military operations analysis (OA) studies by many organizations around the world (McIntosh, 2009). This stochastic agent-based environment uses discrete time steps. In MANA, an abstraction of an engagement-level scenario can include key physical and behavioral attributes of the military entities modeled. The MANA developers strove to minimize the unnecessary detailed physical features of the military assets being modeled.

MANA has a well-developed and intuitive user interface. The "Edit Squad Properties" window in MANA includes several windows, including: General, Map, Personalities, Tangibles, Sensors, Weapons, Intra Sqd SA, Inter Sqd SA, and Advanced. All of these windows allow the user to enter detailed characteristics of the battlefield and military assets being modeled. For instance, Figure 5 is the "Personalities" window, through which users can define the behaviors of the military assets when they interact with other entities, such as enemy, friend, neutral, etc. For example, if a positive number is selected for any enemy scroll bar, it means that the agent will tend to go towards a detected enemy. If the value is negative, the agent will try to move away from an enemy. The magnitude of the value determines how strong this desire is. MANA uses heuristics to weigh the multiple goals of an agent in deciding where to move, who to shoot at, etc. These input characteristics allow the user to control agents' behaviors in missions by giving them values that determine how they behave and interact with other agents and the environment. Additionally, target prioritization, search patterns, rules of engagement, etc., can be determined by assigning input values according to their importance for the mission.

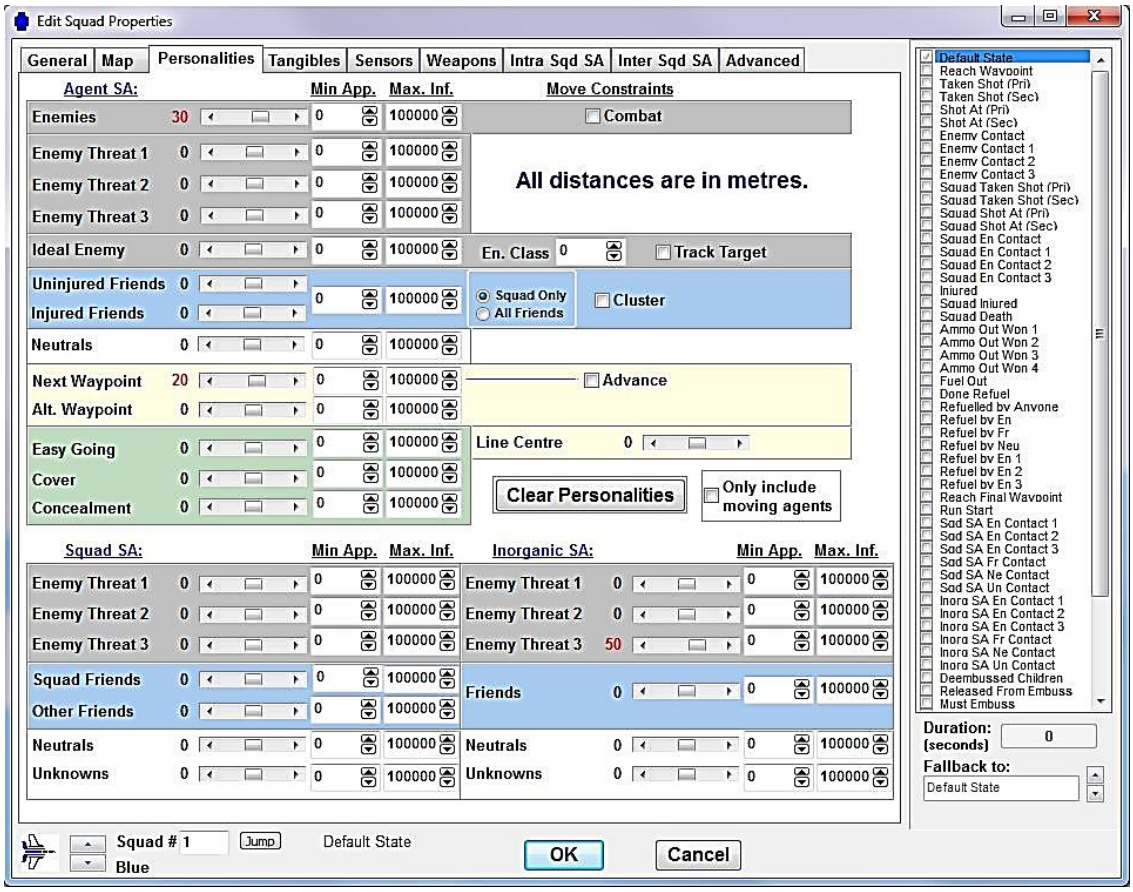


Figure 5. MANA Personalities Window

Figure 6 is a screen shot of the “Sensors” window that allows the user to enter sensor ranges and probabilities for different sensor types, e.g., radars. Also, there are detailed settings, such as detect, classify, and target classes if the advanced mode is selected. The “Tangible” input includes the settings of agents’ physical attributes, such as maximum speed or armor thickness.

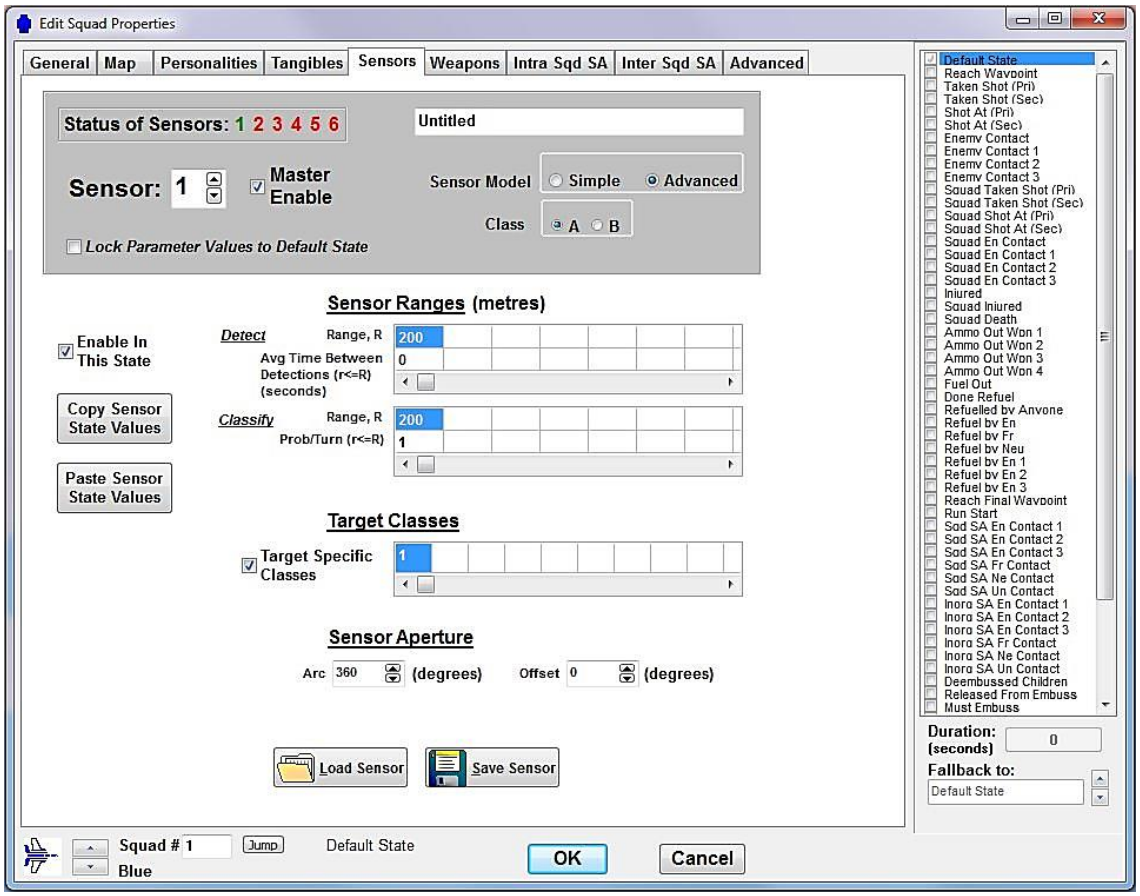


Figure 6. MANA Sensors Window

The column on the far right of the window is called “Trigger States.” It allows the user to change the behavior of military assets in different states, such as “Reach Waypoint,” “Enemy Contact,” “Squad Death,” “Ammo Out,” etc. This allows the agents to change their physical and/or behavior properties according to user defined events, or “triggers.” When one of the trigger state events occurs, a military asset gets new settings and behaviors, as defined by the user. For example, when an Ammo Out condition is triggered, an agent may decide to end his mission and return home.

MANA was selected as our engagement-level simulation for this study because of its utility and ease of use. Moreover, MANA has been extensively used at NPS (see: http://harvest.nps.edu), and additionally, is readily “data farmable”–which means that software exists to run it in parallel over many processors.

MANA provides many options for entering the physical characteristics of military assets, such as sensor capabilities, weapon effectiveness, speed, and fuel capacity—as well as editing battlefield features such as size, terrain type, elevation, etc. The ability to specify the characteristics of military assets, such as fighter aircraft, AWACs, and missile systems makes MANA the ideal stochastic simulation software for this study. However, some of MANA's limitations affected the veracity of our air-to-air mission-level combat model. These include not being able to readily change the probability of hit of a missile for different enemy aircraft speeds and headings. Therefore, the scenario we implement is abstracted from reality. Of course, more complicated mission-level models, such as BRAWLER, are more realistic; however, they are too cumbersome to work with in the time frame of this research.

C. SAMPLING METHODS TO GENERATE INPUTS FOR HIGHER LEVEL MODEL

In our demonstration of hierarchical combat modeling, we use several sampling methods to obtain data from the output data library of the engagement-level model. We briefly discuss those in this section.

1. Using the Overall Sample Mean

A sample mean (average) is a statistic computed from a collection of data and it is the most commonly used estimator of a population mean. Suppose that there are n number of random variables in a sample and $X_1, X_2, \ldots, X_n$ are identically distributed random variables or observations such that they have a finite population mean μ and a finite population variance σ^2. The sample mean ($\bar{X}_{(n)}$)and sample variance (S^2), shown in Equations (2.1) and (2.2), respectively, are typically used to estimate μ and σ^2 (Wackerly, Mendenhall, & Scheaffer, 2007).

$$\bar{X}_{(n)} = \frac{\sum_{i=1}^{n} X_i}{n} \tag{2.1}$$

$$S^2 = \frac{\sum_{i=1}^{n} \left[X_i - \bar{X}_{(n)}\right]^2}{n-1} \tag{2.2}$$

Often, only the sample mean, and not the variance, is fed forward as a single point estimate to a higher-level model. But of course, doing so suppresses much information

about the distribution of outcomes, such as the output measure's range, modality, and skewness.

2. Random Sampling

Random sampling can be accomplished in many ways, for example each data point being equally weighted or not, or sampling "with" or "without replacement." In our application, each member of the population (or data point in the data library) has an equal probability of being chosen. Also, we perform our sampling without replacement. Therefore, once a data point is chosen, it is not chosen again for that particular round of sampling. Additionally, each "subject" (data point) is chosen independently from the other elements of the population data. That is, choosing one data point for use does not make it any more or less likely for any other member of the population to be chosen.

D. CAMPAIGN LEVEL MODELING TOOLS

1. Lanchester's Laws

As briefly mentioned in the first chapter, Frederick Lanchester developed paired differential equations for modeling the attrition in combat (with an eye towards aerial combat) during World War I (Lanchester, 1916). His square law equation models losses over time as a function of the sizes of the forces and the rates of attrition inflicted by an individual unit. Lanchester postulated that this modeled "modern combat" and this variant is often applied to aimed fire situations. Lanchester argued that his other model, the linear law, characterized "ancient combat" contextually, he envisioned this as a series of one-on-one duels with a constant loss exchange ratio. There is also an area fire interpretation to the linear law.

The square law models an aimed fire campaign of two forces. Red forces fight Blue forces, and equations 2.3 and 2.4 define Red's instantaneous loss-rate as being proportional to Blue numbers and vice versa. The Red force size is denoted by x and the Blue force size is denoted by y, while a is the attrition coefficient of Blue (i.e., the rate at which one Blue attrits Red) and b is the attrition coefficient of Red.

$$\frac{dx}{dt} = -ay \tag{2.3}$$

$$\frac{dy}{dt} = -bx \tag{2.4}$$

The linear law model represents a series of independent one-on-one duels, as with ancient swordsmen, or an area fire battle. The losses of each force are proportional to both the number of attackers (other side's force size) and the number of targets (their own force size). Linear law instantaneous loss-rates are shown in equations 2.5 and 2.6.

$$\frac{dx}{dt} = -axy \tag{2.5}$$

$$\frac{dy}{dt} = -bxy \tag{2.6}$$

In the research that this thesis builds upon, Pav chose to implement a square law in the campaign-level model of his model hierarchy. In contrast, in this study, the campaign-level air combat model implements a stochastic extension of Lanchester's linear law. The decision of which Lanchester law should be used was made based on the aforementioned research of three academicians at the University of York, which indicated that the linear law is more appropriate.

In order to calculate the attrition coefficients for each side, we use equations 2.7 and 2.8.

$$a = \frac{x\ casualties}{(time)\cdot(one\ x\ participant)\cdot(one\ y\ participant)} \tag{2.7}$$

$$b = \frac{y\ casualties}{(time)\cdot(one\ x\ participant)\cdot(one\ y\ participant)} \tag{2.8}$$

In addition to the force sizes and attrition coefficients, the stochastic model has an uncertain time element. The time that passes until the next casualty comes from an exponential distribution with rate λ (Washburn and Kress, 2009). In a stochastic Lanchester linear model, this rate for each force is calculated by multiplying the force size of Red (x) and Blue (y) by the related force's attrition coefficient (a or b), which is shown in equations 2.9 and 2.10.

$$\lambda_{RED} = axy \tag{2.9}$$

$$\lambda_{BLUE} = bxy \tag{2.10}$$

Since the time to the next casualty is assumed that the minimum of two independent exponential distributions, the expected time to the next casualty can be calculated using equation 2.11. In our model, we calculate the time until the next casualty based on a random draw from an exponential distribution. The probability that the next casualty suffered will be a Red is determined with equation 2.12. We note that, unconventionally, we choose Red to be "our" side, the "good" side.

$$E[T|x,y] = \frac{1}{\lambda_{RED}+\lambda_{BLUE}} \tag{2.11}$$

$$P[X|x,y] = \frac{\lambda_{RED}}{\lambda_{RED}+\lambda_{BLUE}} \tag{2.12}$$

Employing the equations given in this section, an air campaign between Red and Blue, implementing the stochastic extension of Lanchester's linear law model, is instantiated using the programing language PYTHON 2.7.

2. Python 2.7

The programming language Python, version 2.7, was used to implement the aforementioned stochastic Lanchester linear law campaign model. Python 2.7 is an open source language that includes many analytical packages and tools that vary according to the purpose of the user. It can be freely downloaded from the http://www.python.org. Python 2.7 is a powerful, versatile, general-purpose and dynamic open-source coding language that provides a wide range of available functions and packages. Also, the author utilizes experience in coding in the IPython Notebook, which is an interactive computational platform of Python, where the programmer can read/write data easily, run the code interactively block by block, and add text, plots, etc., as desired. Python is deemed by many to be easy to read, write, and understand.

E. ANALYSIS TOOLS

1. Metamodeling

Understanding the relationship between input factors and the outputs of a model is essential in simulation modeling. One way to quantify the relationship is through a metamodel. A metamodel is a "model of a model." For example, a relatively simple

regression equation may be used to approximate the relationship between the inputs and outputs of a complex simulation. Meta-modeling provides the analyst with a surrogate model that is much more intuitive and fast-running than the original underlying model or dataset. Used properly, in some situations, metamodels can prevent an analyst from having to run time-consuming simulations. Various types of metamodels are used in simulation analysis, including regression equations, partition trees, and Gaussian process models (Barton, 1998).

2. JMP

JMP Pro 12 is used as the statistical analysis tool for this study. It is a computer program created for data visualization and statistics, developed by a unit of the SAS Institute. JMP is widely used by analysts in the applications of design of experiments and scientific research. JMP allows users to interactively manipulate and investigate their data without writing code. In addition to easy data visualization, this program allows the user to utilize many powerful visual and statistical tools, such dynamically-linked plots and graphs, histograms and summary statistics, regression models, partition trees, and many others. More information about JMP can be obtained at http://www.jmp.com/.

3. Distributed High Performance Computing (HPC)

In order to execute the designs of experiment used in this study, we utilized the high-performance computing cluster, consisting of 160 processors, owned and maintained by the Simulation Experiments and Efficient Designs (SEED) Center of the Operations Research Department at the Naval Postgraduate School. The cluster has software installed that automatically generates and manages the running of individual design points of an experimental design. Additionally, post-processing software has been written to manage the collection and summarization of individual simulation run outputs into one file convenient for analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SCENARIO AND MODEL DESCRIPTION

In this chapter we describe the scenarios that were employed in our study of hierarchal combat modeling, including the limitations and capabilities of the software programs used in the modeling. The scenarios give context to the models developed. In the last part of this chapter, our high-resolution and low-resolution models are explained and illustrated with diagrams.

A. SCENARIO DEVELOPMENT PROCESS

In this study, one country, herein referred as Red, has a potential future air campaign against a fictional enemy country, herein referred to as Blue. Recall that Red is "our" side. Red is a Western country and her Air Force uses Western Bloc–type aircraft, radars, and weapon systems; whereas Blue's Air Force uses Eastern Bloc equipment. In more detail, Red has fighter aircraft FX, airborne warning and control system aircraft AWACS, and Blue has fighter aircraft MY and a stationary radar system. Red FXs carry advanced medium-range air-to-air radar guided missiles, which we will call X-MRGM. On the other hand, Blue MYs carry Y-MRGM missiles, which have similar features to X-MRGM and are made by an Eastern Bloc country.

Due to the fact that the main purpose of this research is to analyze the propagation of error within hierarchal air combat models, a basic scenario was generated. In our experimental study, our context is that Red is considering investments in air combat technologies, such as aircraft, radars, and weapon systems. The simplicity of the given scenario and the models generated will hopefully allow us to focus on error propagation and facilitate other analysts in using the models of this study in future studies.

B. CONTEXT FOR THE SCENARIO

A Red reconnaissance jet FX was intercepted and shot down by a country Blue's surface to air missile (SAM) in international airspace. Two Red jet pilots were killed. The incident was a part of a series of incidents between Red and Blue since the beginning of

Blue's independence war, and this incident escalated the tensions between the two countries.

After the Red jet was shot down, Red adopted new rules of engagement. In the following days, Red's Air Force shot down one MY and one Blue drone after they violated Red's airspace. Blue's Air Force then procured 100 new MY fighter aircraft from country yellow. MY has similar features to Red's FX aircraft.

A couple of months later, two Red FXs take off from an air based located in the south east of the country. The Red FXs are executing a combat air patrol (CAP) close to the border with Blue when one of Blue's SAM launchers fires a missile targeting Red FXs. The FXs dropped flares and manage to get rid of the incoming SAM missile. Afterwards, to prevent further missile attacks, the Red government decides to retaliate with an effective air strike on the SAM missile launcher located in the northwest part of Blue. Therefore, the Red Air Force prepares an air-to-ground attack plan. Based on their intelligence, Red headquarters learns that the Blue Air Force will defend the SAM missile launcher with a new MY squadron located very close to the launcher.

In order to attack the SAM launcher, Red's Air Force forms an air strike package, including bomber aircraft, 25 FX fighter jets (for possible air-to-air engagements), one AWACS (for pre-raid and post-raid reconnaissance) and one tanker aircraft (to extend the mission radius). The FXs are equipped with air-to-air missiles X-MRGMs, and will fly to the operation area. On the other hand, Blue has weaker intelligence that there may be a Red air raid. Therefore, he has 25 MY aircrafts on scramble mode in the closest base to its border with Red. Those MYs are loaded with Y-MRGM missiles.

The Red AWACs has a 360° view of the horizon. Blue's stationary radar can see if any Red aircraft flies towards Blue. Both radars can simultaneously detect and track multiple air targets.

Like every other simulation model, the model created for this study cannot capture all real-world conditions, behaviors, and characteristics of military assets. Some of the more important limitations, constraints, and assumptions are discussed in the following paragraphs. Based on our contextual scenario, to see the engagement

performance of FX and MX, an engagement-level model was generated in MANA. In the engagement-level model, two versus two air-to-air engagements are simulated in order to generate a data library for the campaign model. Later, using outputs of the MANA model, attrition coefficients for each aircraft type are calculated. These attrition coefficients are then used in a stochastic linear Lanchester model as a campaign-level model with 100 Red aircraft versus 100 Blue aircraft engaging each other over four time periods.

C. LIMITATIONS AND ASSUMPTIONS

First, accessing real performance characteristics of military assets, such as radar capabilities and missile ranges, is impossible without making this research classified. Since our focus is on error propagation, actual data is not needed. Therefore, IHS JANE's web (http://www.janes.com/) database is used to generate plausible and reasonable aircraft, radar, and missile performance characteristics. Again, the focus here is on the methodology not on specific weapon systems, and keeping the study unclassified, so other researchers will be able to use this paper in their future studies.

In addition to using unclassified performance characteristics, the constraints of MANA resulted in a simpler and less detailed scenario than could have been constructed with a higher resolution model.

(1) Assumptions and Limitations for the Engagement-Level MANA Model

1. Before making the decision of how many aircrafts to include in our high-resolution MANA model, discussions with three fighter aircraft pilots and open Internet research about air combat operations were undertaken by the author. Both the discussions and air combat history showed that fighter aircraft do not fly alone. Therefore, instead of one-on-one engagements, two-versus-two aircraft battles were selected for the MANA model.
2. One of the major factors that impact the probability of hit and effective range of an air-to-air missile is the geometry (i.e., routes and directions) of the engaging aircraft. A missile reaches its maximum effective range if a head-to-head engagement occurs, other than that; the effective maximum range significantly decreases. The probability of hit and the effective range of a missile are not readily modeled in MANA as a function of the aircrafts' orientations and kinematics.

3. In MANA, the agents cannot explicitly exercise fire control over wire-guided munitions (Pav, 2015) and missile flying time is neglected. Therefore, guided weapons, such as a missile, were modeled as a bullet that has probability of hit that varies according to range, but not aspect or heading.

4. Each aircraft is assumed to have unlimited air fueling and missiles.

5. The aircraft of each force carry one type of missile, but differ according to the force type. This holds for the radars as well.

6. Tactical deployment of aircrafts is minimal since air combat tactical formations are difficult to model in MANA.

(2) Assumptions and Limitations for the Campaign-Level Model

1. It is assumed that modern aircraft can engage more than one aircraft.

2. Varying Red and Blue force sizes while also varying engineering factors would increase the amount of work and analysis, since it would add two more dimensions to our model inputs and consequently to our data library. Thus, we keep each side with a fixed number of aircraft (100).

3. The history of air combat operations shows that air operations can be executed more than once a day or once in more than a day. For instance, the Falkland war between Argentina and the United Kingdom over two British overseas territories took ten weeks—with many days not involving air battles (Chant, C., 2001). On the other hand, air operation time periods can be executed sequentially in a day, like in surge operations early in the Iraq War by the U.S. Forces (Pape, 2004). Additionally, a discussion about air combat operations with Tom Lucas and Wayne P. Hughes was made by the author before making the decision of what time element to use in attrition coefficient calculations (T.W. Lucas & W.P. Hughes, personal communication, March 10, 2016). Based on historical experience and the discussions, to calculate attrition coefficients, a single air operation time is used as a period—with four periods modeled.

4. All forces are homogenous.

5. Neither force fights to the death. Each side has a breakpoint (tolerance for loss), beyond which they choose to terminate battle for that time period or campaign.

6. Precise tactical deployments are not used.

7. Only force size and attrition coefficients are provided numerically to predict the outcome and number of casualties of each side at the end of the campaign.

D. HIERARCHICAL MODELS OF APPLICABLE SCENARIO

Based on the applicable scenario, modeling and sampling tools were employed to generate a hierarchal air combat analysis including a high-resolution (engagement-level MANA model) and a low-resolution (campaign-level stochastic Lanchester linear model) model. The models that were used in this study are depicted in Figure 7 as a hierarchy pyramid.

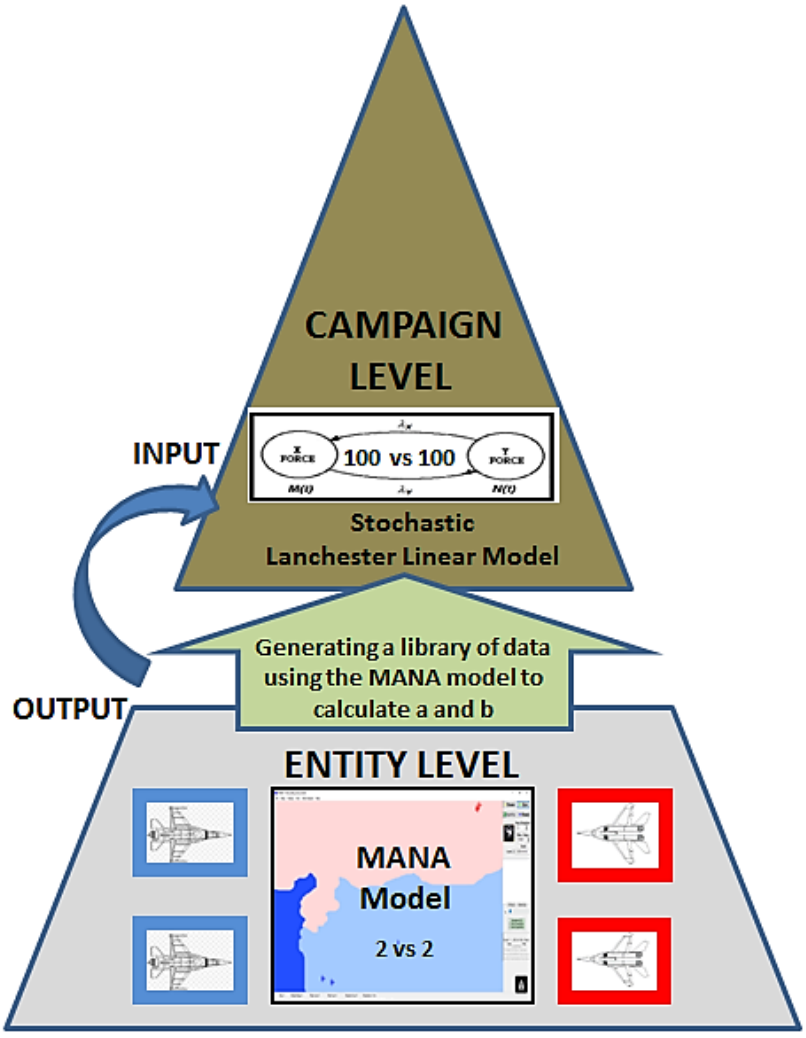


Figure 7. Hierarchical Air Combat Models of This Study

The two models shown in Figure 7 constitute the hierarchal air combat model family we will experiment with. The measures of effectiveness (MOEs) of the engagement-level model are used to generate measures of performance (MOPs) input for the campaign-level model by utilizing various sampling methods. The MOEs and MOPs for each level model are shown in Table 1.

Table 1. MOPs and MOEs of Each Model Level.

MODEL LEVEL	MODEL TOOL	MOPs	MOEs
ENGAGEMENT	MANA	- AIRCRAFT PERFORMANCE CHARACTERISTICS GENERATED FROM NOLH AND R5FF DOEs.	- WINNER(THE SIDE WITH FEWER CASUALTIES) - # of LOSSES
CAMPAIGN	STOCHASTIC LANCHESTER LINEAR MODEL	- FORCE SIZE (CONSTANT AND GIVEN) - ATTRITION COEFFICIENTS CALCULATED FROM MANA MODEL'S # OF CASUALTIES	- P(Win) - RED LOSSES

Table 1 briefly shows the relationship between the MOEs of the engagement-level model and the MOPs of the campaign-level model. Since performance characteristics are varied by utilizing DOE, effects of those changes on campaign MOEs will be analyzed. The probability of winning is denoted as P(Win). This provides the ultimate measure of success, while the expected number of casualties in victories quantifies some of the uncertainty of campaign outcomes that creates risk for decision makers.

(1) High Resolution – Engagement-Level Model

As previously mentioned, the engagement-level model of a two versus two air engagement is developed in MANA. The simulated battlefield is 480 by 500 miles. Two Red fighter jets and one Red AWACs take off from the southern part of the Red country in order to escort Red bombing jets whose goal to bomb a SAM launcher in the western part of the Blue country. If any Red fighter survives at the end of an engagement, it returns to base. Red fighters have 100% reliable communication with the AWACS and the AWACS provides improved radar support to detect enemy aircraft and transfer enemy aircraft coordinates to the Red jets. The AWACS and fighter jets fly with the same speed of 400 kilometers per hour; however, the AWACS flies at a very high altitude,

which allows him to stay away from any engagement with hostile aircraft. After the Red AWACS detects enemy aircrafts, the Red fighters' speed increases from 400 kilometers per hour to 800 kilometers per hour to engage enemy combatants as soon as possible. On the other hand, Blue fighter pilots are waiting on the ground for scrambling. After the stationary radar located in northern side of Blue country detects enemy aircraft and sends information about them, two blue pilots will quickly get into their fighters and move to intercept the intruders with a speed of 800 kilometers per hour. Each two-vs-two engagement starts with the take-off of the Red fighters and ends with the landing of the last surviving aircraft after shooting down the both fighters of other side. The length of engagement was selected based on the discussion with Lucas and Hughes (T.W. Lucas & W.P. Hughes, personal communication, March 10, 2016). See Figure 8 for an annotated MANA screen shot of the model.

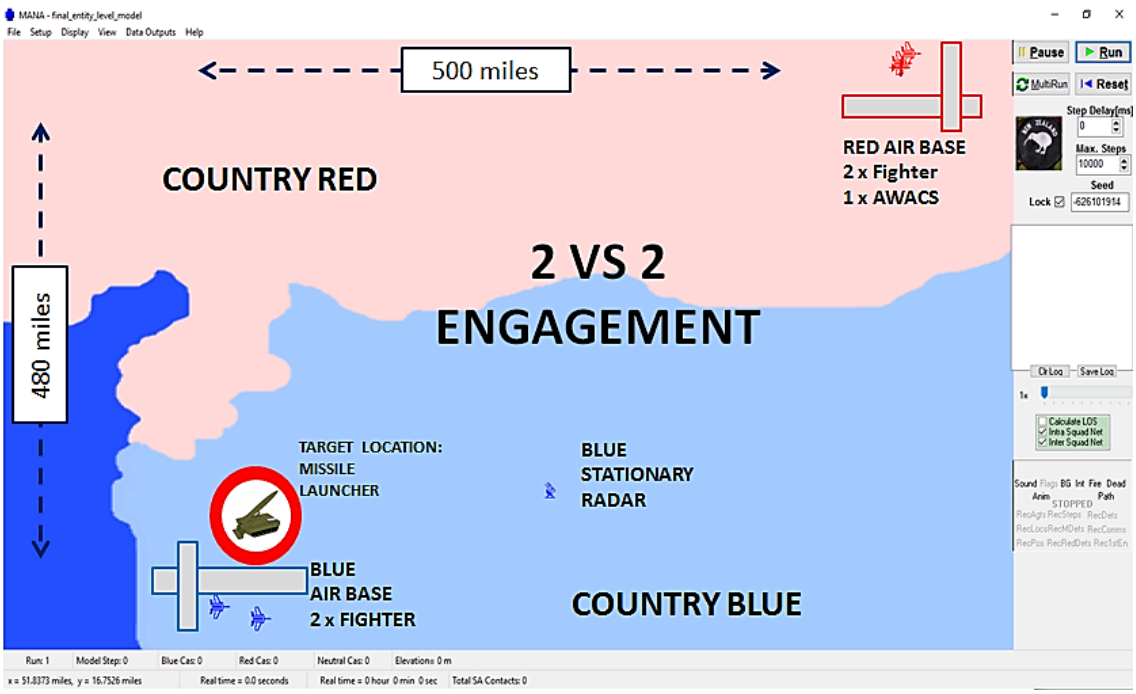


Figure 8. High Resolution Engagement-Level MANA Model

(2) Low Resolution – Campaign Level Model

As previously mentioned, this thesis employs a stochastic Lanchester linear model as the campaign model. In our campaign model, Red has 100 aircraft and Blue has 100

aircraft. Since this study's goal is to analyze error propagation within hierarchal air combat models, to keep the campaign model simple, the campaign consists of four time periods that include the following features. Red performs 4 air operations including 25 fighters each time and blue counterpoises these air operations with same number of fighters. The main features of the campaign model are presented below.

- For each of the four time periods, 25 Red will fight 25 Blue.
- Each side will have a break point (tolerable loss) of six remaining aircraft, just under a quarter of the operation's force size. If a break point is reached, that side will disengage and that time period's battle will end.
- At the end of the four time periods, the winner is the side with fewer casualties. There is no tie situation.
- Length of each operation are time between operations are neglected since we are interested in P(Win) and number of casualties.
- The data generated from each run includes: Red casualties, Blue casualties, and the winner of the campaign.

The campaign model of this study is written by the writer as a code in Python 2.7 notebook platform, where the code is presented in appendix section of this paper.

IV. DATA-FARMING, MODEL RUNS, AND ANALYSIS

This thesis examines methods for linking a campaign model's inputs to an engagement model's outputs. Therefore, one of our goals is to make recommendations about how to quantify the uncertainty within campaign model output, given variability in the inputs, which will allow analysts to assess risk to the commander more comprehensively. Two models that together form our hierarchy of models, a two-vs-two engagement level model and a campaign level stochastic Lanchester model are developed and employed.

The overall work flow of our analysis is shown in Figure 9. In steps 1A through 1E, we use two different experimental designs, described in the next section, to perform an experiment on the MANA model. Each MANA replication starts with the take-off of the Red fighters and ends with the landing of the last surviving aircraft after shooting down the both fighters of other side. We then analyze both the full and summarized data. In the summarized data set, we summarize each Design Point (DP) by its mean and standard deviation, over the stochastic replications. The full and summarized data from our MANA experiments form what we will call the MANA library of data. In step 1F, we calculate estimates of *a* and *b* (inputs to the Lanchester model) for each run of both NOLH and R5FF data libraries. Additionally, a column called "Win?" is calculated—and the assigned value is 0 if our side, the "Red" side lost and 1 if our side won. We remind the reader that in our model we take an unconventional approach in deciding "Red" to be the "good guys." The side with fewer casualties is the winner in MANA model. In the summarized data, mean(Win?) is renamed P(Win), our estimate for the probability of winning. In step 1G, using the summarized data, we fit stepwise regression models for *a*, *b*, and P(Win), as a function of the experiment variables. We then compare the NOLH and R5FF results, including the metamodels generated.

In steps 2A through 2C we explore different methods, which we explain in detail in this chapter, to translate the MANA results into inputs needed for the campaign model. Step 2C involves simulating the "direct linking" of the campaign model's inputs to the engagement-level model's outputs.

In step 2D.1, we perform a design of experiment directly on the stochastic Lanchester model, varying *a* and *b* over ranges informed by the MANA library of data. In step 2D.2, we fit a regression model for P(Win) as a function of *a* and *b*.

In step 2D.3, a composite (embedded) metamodel of P(Win) is constructed. We explain the composite (embedded) metamodel in detail in the further part of this chapter. from step 2D.2, we obtain P(Win) as a function of *a* and *b*. And, from step 1G, we obtain *a* and *b* as functions of the experiment variables. So, in this step, we simply substitute *a*=f(experiment variables) and *b*=f(experiment variables) into P(Win)=f(*a*,*b*), obtaining a composite, embedded metamodel P(Win)=f(experiment variables).

In step 3, we compare the composite metamodel approach to the result of 2C, directly linking the Engagement and Campaign Model. And in step 4, we compare the campaign metrics based on the different sampling methods. The next section will describe each of these steps in more detail.

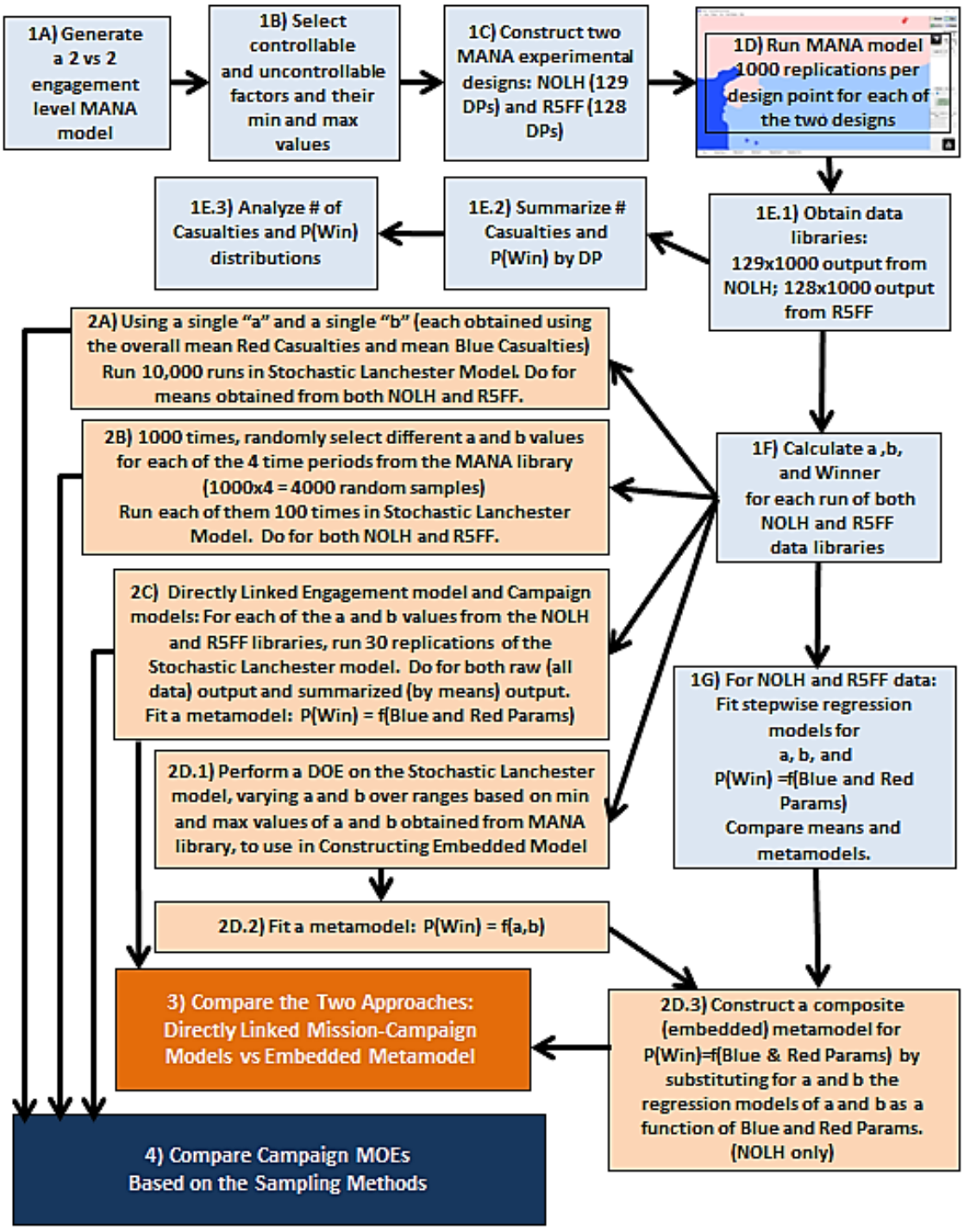


Figure 9. Overall Work Flow Diagram

A. ENGAGEMENT-LEVEL MODEL PROCESS

1. Data Farming (Blocks 1B, 1C of Work Flow Diagram)

The performance characteristics of the aircraft in the model are notional. In the real world, these characteristics would vary depending on enemy counter-measures, aircraft orientation, radar and missile capabilities, operator training, and other aspects of air-to-air combat. In order to explore the impact of aircraft characteristics on performance in the MANA model, we conduct a designed experiment. Data farming is our metaphor for iterated design and analysis of experiments. We use two different experimental designs, so that we may compare and contrast the results. The two designs utilized are the Nearly Orthogonal Latin Hypercube (NOLH) and the Resolution V Fractional Factorial (R5FF). Both of these techniques are relatively efficient, with respect to the number of design points required, as compared to a full-factorial design which tests all possible combinations. One key difference between the two is that the NOLH is a space-filling design that allows flexibility in the analysis (for example, higher order terms can be estimated in a regression model) while the R5FF tests each factor at only two levels (low and high). Therefore, an advantage of the R5FF is that it tests the more extreme "corner points" but does not allow for capturing curvature in the response surface, via estimation of quadratic or higher terms. By employing both experimental design methods, performance characteristics of Red and Blue aircraft were varied around open source (HIS Jane's data library) base case values. A total of ten variables were varied, and these are shown in Table 2.

Another advantage of the NOLH design is that it can be easily developed using a spreadsheet freely available from the SEED Center for Data Farming's website (http://harvest.nps.edu). Analysts have several catalogued designs in the spreadsheet to choose from. The catalogued designs have different numbers of design points (DPs) according to the number of factors desired in the experiment.

Table 2. Red and Blue Aircraft Design Factors in DOE for MANA Model.

FACTORS	DESCRIPTION	LOW	HIGH
RED FIGHTER AIRCRAFT WEAPON EFFECTIVE RANGE (AC-EffRng)	Maximum effective range that a red fighter aircraft missile hits an enemy aircraft. It is varied since an enemy aircraft will have different directions , maneuvours, etc.	50,000 meters	80,000 meters
RED AIRCRAFT WEAPON PROBABILITY OF HIT (AC-EffRng-Phit)	Probability that a missile fired from a red fighter aircraft hits an enemy aircraft. It is varied since enemy aircraft will have counter maneuvers, flares and missile has its own probability of hit.	0.8	0.9
RED AWACS RADAR RANGE (AWACS-Rng)	Maximum range that a red AWACS can detect an enemy fighter aircraft. Variation is generated based on enemy counter measures and radar detection capabilities.	320,000 meters	400,000 meters
RED AWACS PROBABILITY OF CLASSIFICATION (AWACS-Pclass)	Probability that red AWACS classifies correctly an enemy aircraft after AWACS detects that enemy aircraft by its radar. It is varied since enemy will have counter measures such as jammer system etc.	0.8	1
COMMUNICATION LATENCY (CommLat)	Organic communication between red aerial assets may have small amount of time delay in transfering information to each other after radar detecting an enemy aircraft in radar.	0 second	2 seconds
RED AIRCRAFT STEALTH ACStealth	Red fighter aircraft stealth is the amount of protection from view that the entity has, and ranges fron 0 to 100%, where 100% is invisible.It is partial and varied based on the idea that new fighters have partial stealth capabilites.	0%	30%
BLUE AIRCRAFT WEAPON EFFECTIVE RANGE (EnAC-EffRng)	Maximum range that a blue fighter aircraft can fire a missile. It is varied since an enemy aircraft will have different directions , maneuvers, etc.	50,000 meters	80,000 meters
BLUE AIRCRAFT WEAPON PROBABILITY OF HIT (EnAC-EffRng-Phit)	Probability that a missile fired from a red fighter aircraft hits an enemy aircraft. It is varied since enemy aircraft will have counter maneuvers, flares and missile has its own probability of hit.	0.8	0.9
BLUE STATIONARY RADAR RANGE (EnRadarRng)	Maximum range that a Blue stationary radar can detect an enemy fighter aircraft. Variation was genereted based on enemy counter measures and radar detection capabilities.	220,000 meters	280,000 meters
BLUE STATIONARY RADAR PROBABILITY OF CLASSIFICATION (EnRadar-Pclass)	Probability that Blue stationary radar classifies correctly an enemy aircarft after radar detects that enemy aircraft by its radar. It is varied since enemy will have counter measures such as jammer system etc.	0.8	1

Although 65 design points would have been sufficient for our 10 factors, we chose the next larger catalogued design, the one containing 129 design points, in order to fill more of the space, and also to have a number of design points close to the number of design points in the R5FF. The number of DPs required for a R5FF for 10 variables is

128. An advantage of having similar numbers of DPs for the two designs is so that statistical power (the ability to detect a statistically significant effect, if one is present) should be roughly equal (for main effect models—with an advantage to the optimal R5FF). This is advantageous so that the metamodels created from each design's output are more directly comparable.

2. Design Point Analysis

In order to generate a scatter plot of the experiment factors and generate a matrix of pairwise correlations to analyze design points, we conduct a multivariate analysis in JMP. The results, for the NOLH design and the R5FF designs, are shown in Figure 10 and Figure 11, respectively.

Engagement Model NOLH Output Appended Summary - Multivariate Analysis

Correlations

	AWACS-Rng	AWACS-Pclass	CommLat	AC-EffRng	AC-EffRng-Phit	ACStealth	EnAC-EffRng	EnAC-EffRng-Phit	EnRadarRng	EnRadar-Pclass
AWACS-Rng	1.0000	-0.0013	-0.0049	-0.0000	-0.0155	0.0056	-0.0043	-0.0018	0.0008	-0.0070
AWACS-Pclass	-0.0013	1.0000	-0.0000	-0.0006	-0.0000	0.0012	-0.0058	-0.0041	-0.0061	-0.0203
CommLat	-0.0049	-0.0000	1.0000	-0.0061	0.0320	0.0056	0.0056	0.0054	-0.0008	-0.0064
AC-EffRng	-0.0000	-0.0006	-0.0061	1.0000	0.0163	0.0063	-0.0016	0.0221	0.0025	0.0006
AC-EffRng-Phit	-0.0155	-0.0000	0.0320	0.0163	1.0000	-0.0099	-0.0217	0.0134	0.0152	-0.0108
ACStealth	0.0056	0.0012	0.0056	0.0063	-0.0099	1.0000	-0.0028	0.0244	0.0007	-0.0021
EnAC-EffRng	-0.0043	-0.0058	0.0056	-0.0016	-0.0217	-0.0028	1.0000	-0.0006	0.0000	-0.0066
EnAC-EffRng-Phit	-0.0018	-0.0041	0.0054	0.0221	0.0134	0.0244	-0.0006	1.0000	0.0327	0.0022
EnRadarRng	0.0008	-0.0061	-0.0008	0.0025	0.0152	0.0007	0.0000	0.0327	1.0000	0.0108
EnRadar-Pclass	-0.0070	-0.0203	-0.0064	0.0006	-0.0108	-0.0021	-0.0066	0.0022	0.0108	1.0000

Scatterplot Matrix

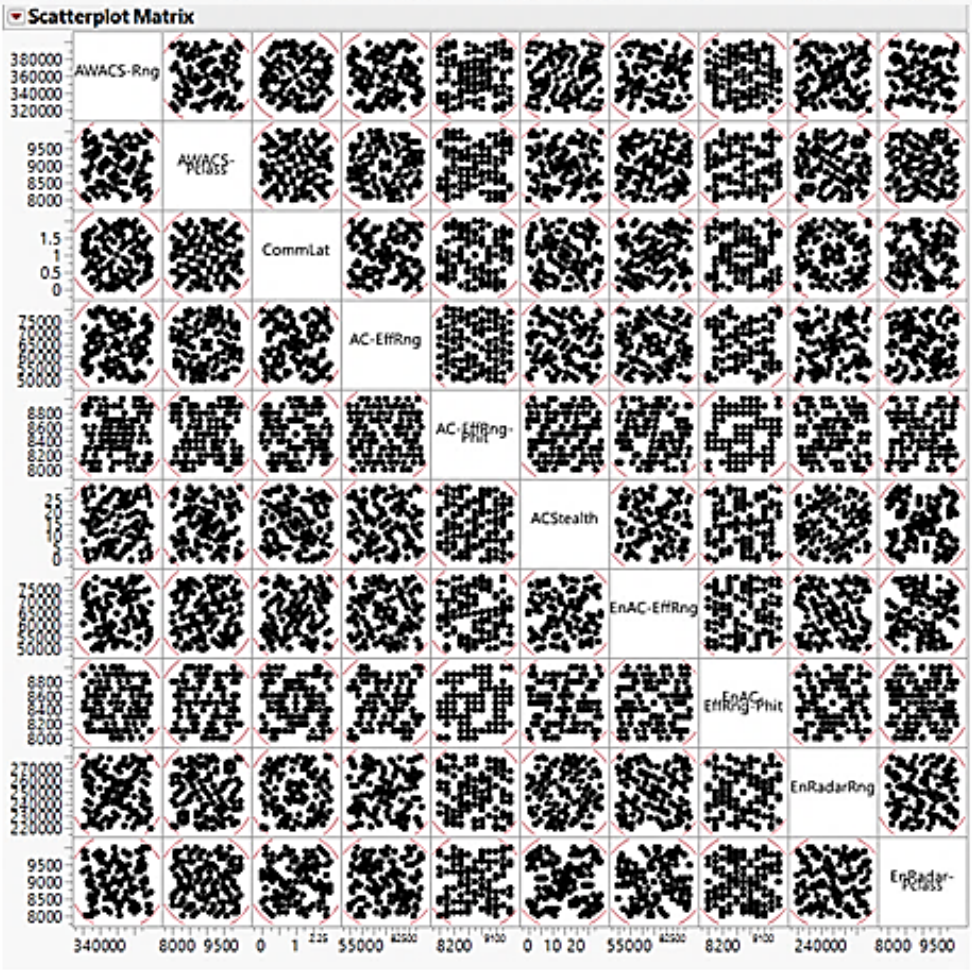


Figure 10. Multivariate Analysis of NOLH Design Points

Figure 10 illustrates the space-filling aspect of the NOLH design, which allows us to explore the interior of the experimental region as well as remain fairly flexible with our analysis goals. As previously mentioned, with the output produced from the NOLH design, we can estimate higher order terms in regression models, such as quadratic effects. We see from the correlation matrix that all pairwise correlations between

variables are below the +/- 0.03 threshold which satisfy the near-orthogonality constraint (Cioppa & Lucas, 2007).

Engagement Model R5FF Output Appended Summary - Multivariate Analysis

Correlations

	AWACS-Rng	AWACS-Pclass	CommLat	AC-EffRng	AC-EffRng-Phit	ACStealth	EnAC-EffRng	EnAC-EffRng-Phit	EnRadarRng	EnRadar-Pclass
AWACS-Rng	1.0000	0.0000	-0.0000	0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000	0.0000
AWACS-Pclass	0.0000	1.0000	0.0000	-0.0000	0.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000
CommLat	-0.0000	0.0000	1.0000	0.0000	-0.0000	0.0000	-0.0000	-0.0000	-0.0000	0.0000
AC-EffRng	0.0000	-0.0000	0.0000	1.0000	-0.0000	0.0000	-0.0000	-0.0000	-0.0000	-0.0000
AC-EffRng-Phit	0.0000	0.0000	-0.0000	-0.0000	1.0000	0.0000	-0.0000	0.0000	-0.0000	0.0000
ACStealth	-0.0000	0.0000	0.0000	0.0000	0.0000	1.0000	-0.0000	-0.0000	-0.0000	0.0000
EnAC-EffRng	0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	1.0000	0.0000	-0.0000	-0.0000
EnAC-EffRng-Phit	-0.0000	0.0000	-0.0000	-0.0000	0.0000	-0.0000	0.0000	1.0000	-0.0000	0.0000
EnRadarRng	0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	-0.0000	1.0000	0.0000
EnRadar-Pclass	0.0000	0.0000	0.0000	-0.0000	0.0000	0.0000	-0.0000	0.0000	0.0000	1.0000

Scatterplot Matrix

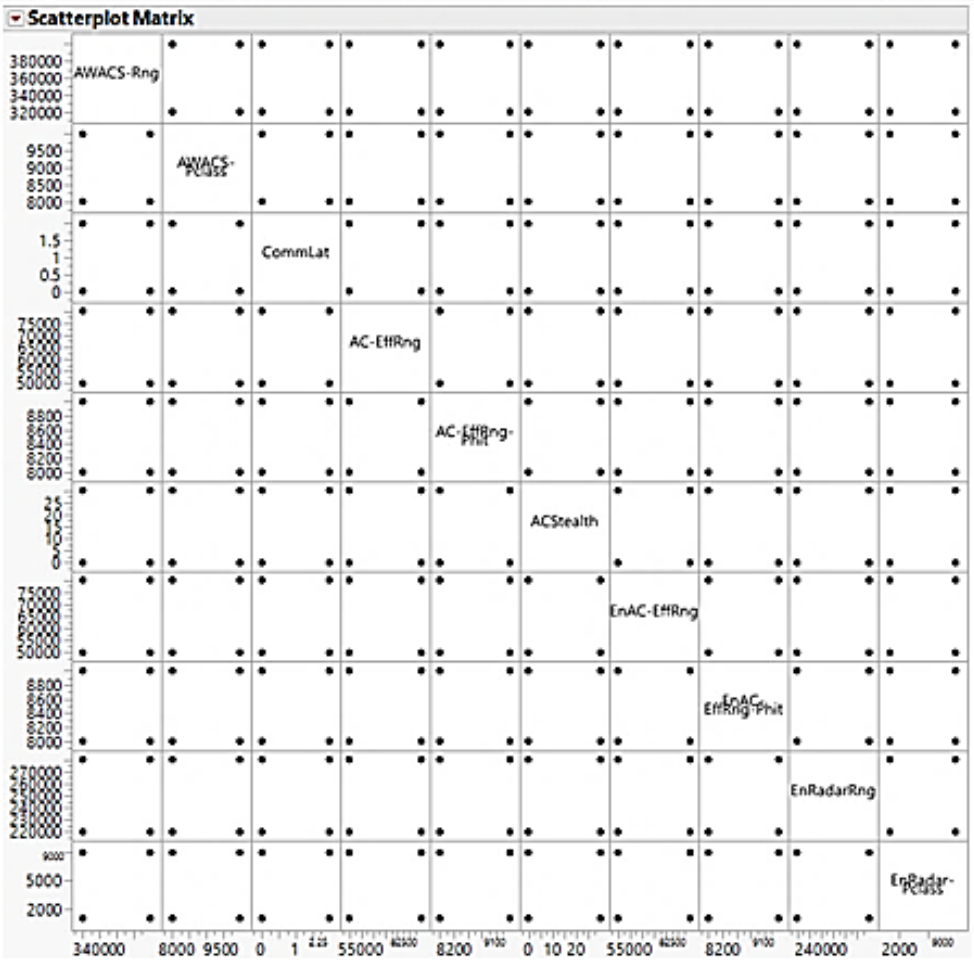


Figure 11. Multivariate Analysis of R5FF Design Points

In Figure 11, it is apparent that the R5FF design samples only at the corners of the input design space and also that all pairwise correlations between factors are equal to 0.

With the R5FF, all two-way interaction terms in regression models can be estimated; however, quadratic effects cannot be estimated because each factor is sampled at only two levels.

3. Engagement Model Runs and Output Analysis (Blocks 1D, 1E of Work Flow Diagram)

For both the NOLH and R5FF experiments, each DP was replicated 1,000 times in MANA. Thus, the NOLH experiment consisted of 129,000 runs while the R5FF experiment consisted of 128,000 runs. As mentioned before, these runs were generated in order to provide a library of data to use as input to the campaign model. In order to conduct data analysis and build metamodels, we use the JMP statistical software.

a. Raw Data Analysis

We start by generating histograms and summary statistics for Blue Casualties, Red Casualties, and Length of Engagement based on the NOLH and R5FF raw (full output) data libraries. The NOLH results are displayed in Figure 12 and the R5FF results are displayed in Figure 13.

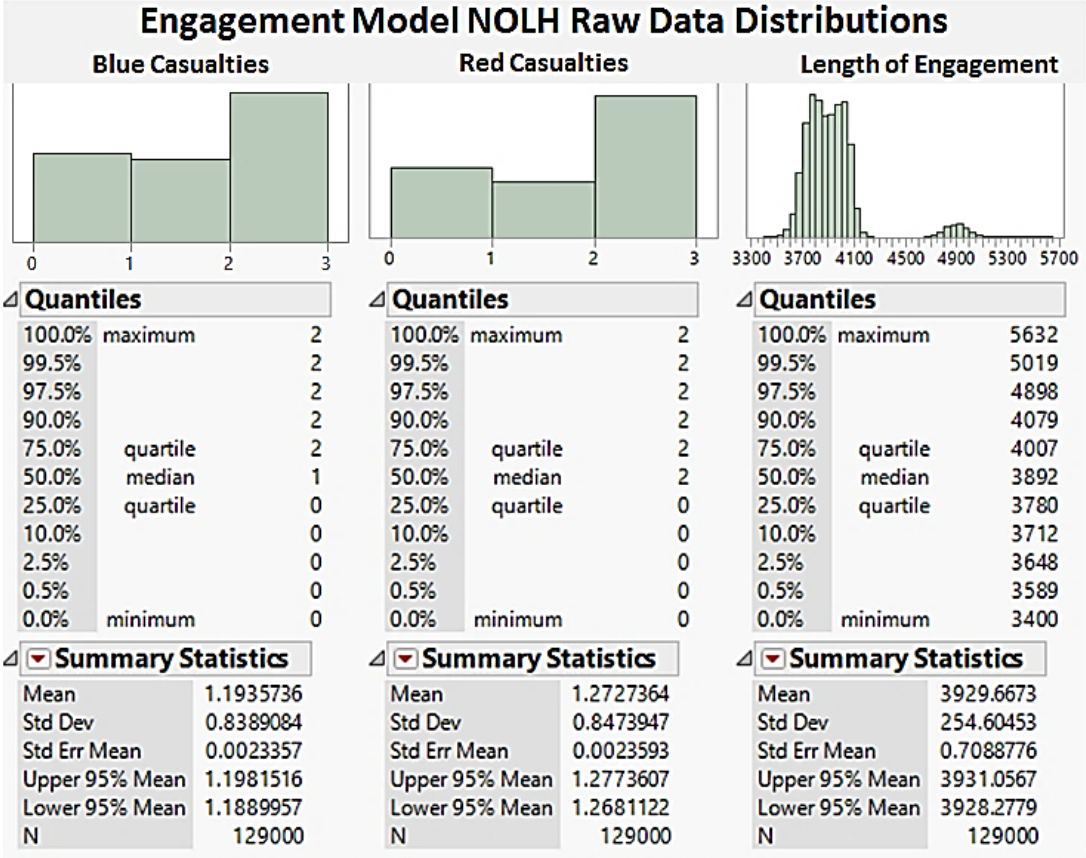


Figure 12. Engagement Model NOLH Raw Output Distributions

The results in Figure 12 show that the mean of Red casualties is larger than the mean of Blue casualties, and that they have very similar standard deviations. We also note that most of the runs end with two casualties for one of the sides, as expected. We also see that the "Length of Engagement" distribution is bimodal. The bimodality is because the higher-valued mode is associated with Red winning and the lower-valued mode is associated with Blue winning. This result is due to the fact that the distance between the Red air base and approximate engagement area is larger than the distance from the Blue air base to the same engagement area. Therefore, if Red wins, it takes more time for them to return their base, and it was decided that this time should be factored into the inputs for the campaign model.

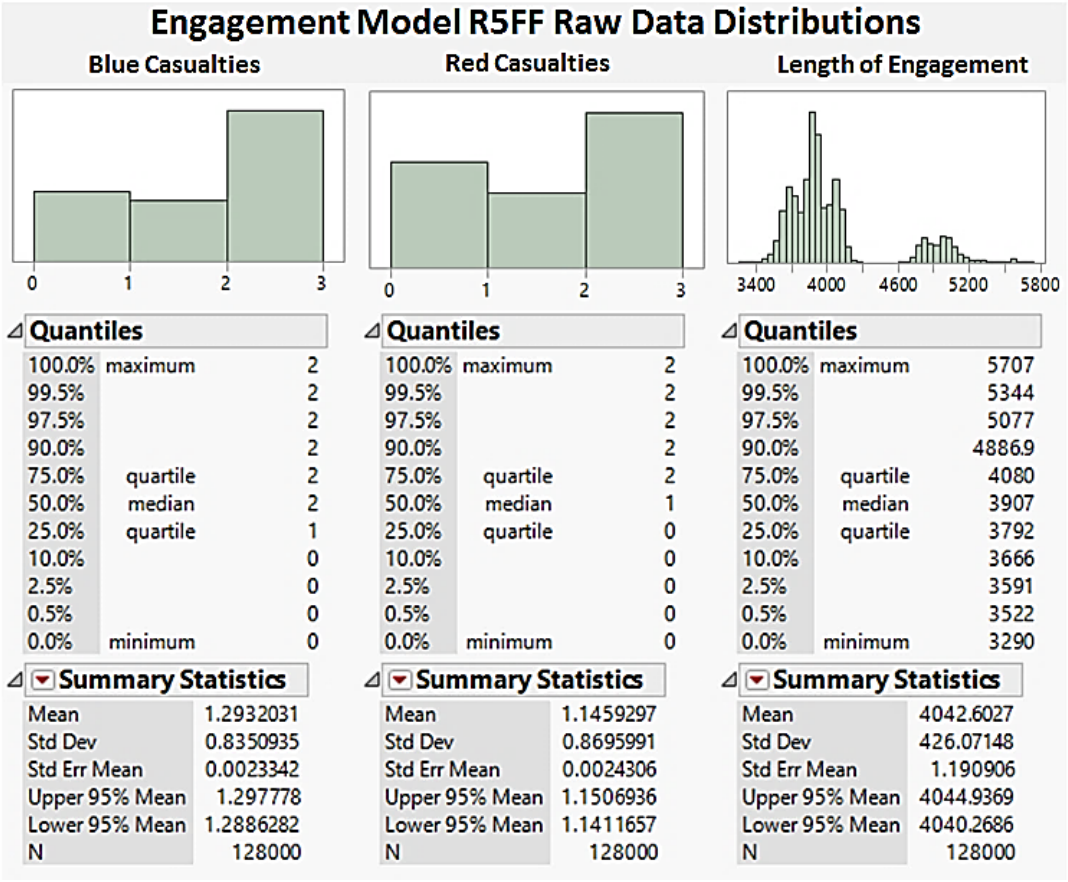


Figure 13. Engagement Model R5FF Raw Output Distributions

The R5FF data histograms and summary statistics share similar features as the NOLH raw data. However, in this case, there are, on average, more Blue casualties than Red casualties. The reason for the different result is likely due to the fact that the R5FF samples only at the more extreme "corner" points.

b. Summarized Data Analysis

We next summarized the raw (full) output data set by calculating the mean of Red Casualties, mean of Blue Casualties, mean of Length of Engagement, and P(Win), over the 1000 replications, for each DP. Therefore, N=129 for the summarized NOLH data and N=128 for the summarized R5FF data. We display the results in Figures 14 and 15.

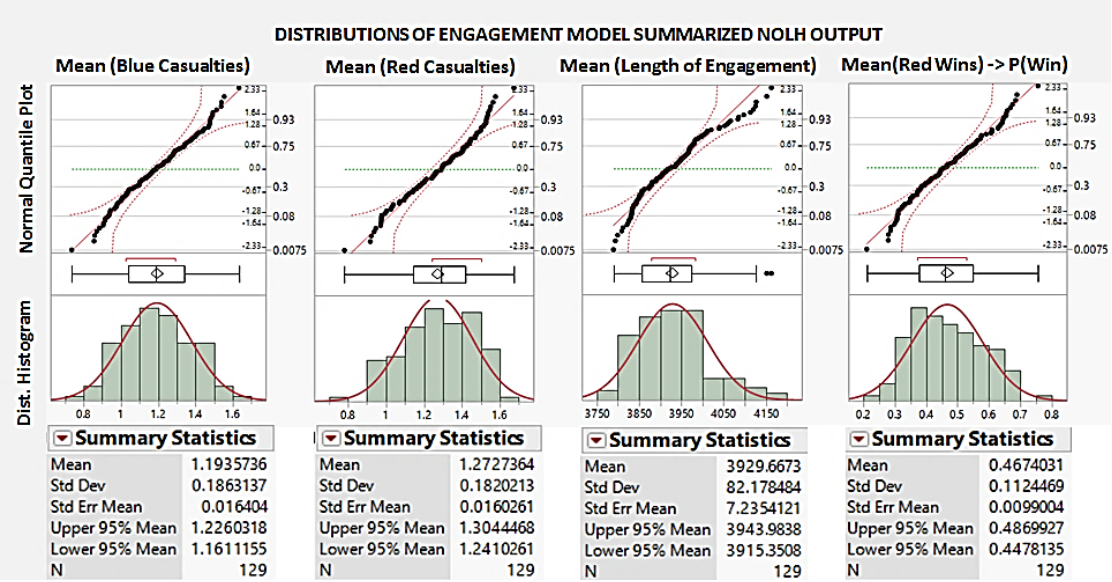


Figure 14. Distributions of Engagement Model Summarized NOLH Output

By visual inspection, the histograms look like they do not stray far from normality. However, in order to assess whether the underlying data is normally distributed more accurately and effectively, we utilize JMP's ability to generate a normal quantile plot and conduct a goodness of fit test.

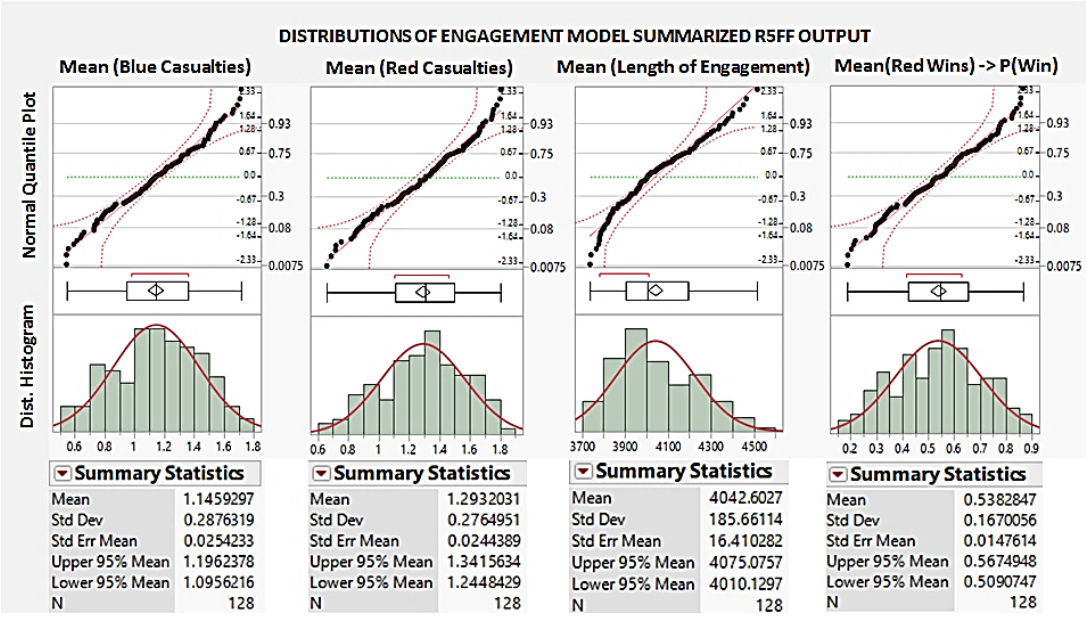
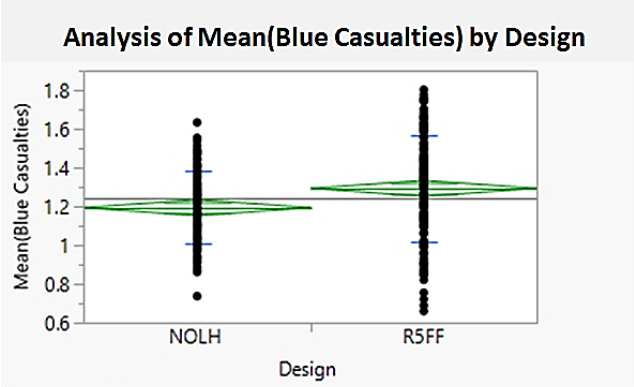


Figure 15. Distributions of Engagement Model Summarized R5FF Output

The normal quantile plot for each of the outputs appear in Figures 14 and 15. Points falling far off the diagonal indicate that they are in a different location than would be expected from a normal distribution (Wackerly, Mendenhall, & Scheaffer, 2007). It is seen that most of the points in these normal quantile plots do fall close to the diagonal lines. We next performed JMP's Goodness of Fit test, which utilizes the Shapiro-Wilk W test (Wackerly, Mendenhall, & Scheaffer, 2007). The null hypothesis for this goodness of fit test is that the data were drawn from a normal population. Rejecting the null would then indicate that we have evidence that the data do not come from a normal population. We do not show here the results for the goodness of fit test, but note that, except for the Length of Engagement distributions (for both data sets), all of the p-values were larger than our chosen 0.05 level of significance, therefore, we do not reject the null hypothesis that that these NOLH and R5FF summarized data are normally distributed. The lack of normality for the Length of Engagement distributions is not surprising since we already realized from visual inspection that the Length of Engagement distributions were bimodal.

Finally, we compare the means of key metrics by design type (group), in order to determine if there is a statistically significant difference between the two groups.



Welch's Test

Welch Anova testing Means Equal, allowing Std Devs Not Equal

F Ratio	DFNum	DFDen	Prob > F
11.4573	1	222.42	0.0008*

t Test
3.3849

Figure 16. Analysis of Means (Blue Casualties) by Design

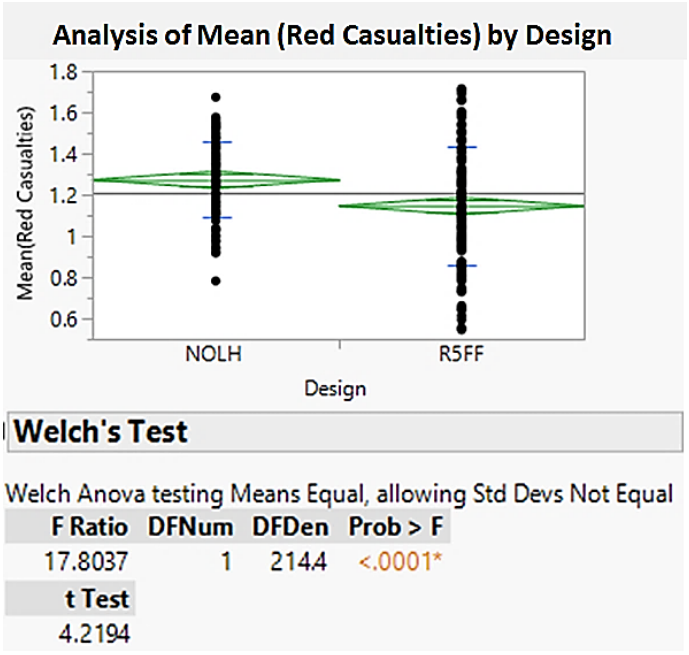


Figure 17. Analysis of Means (Red Casualties) by Design

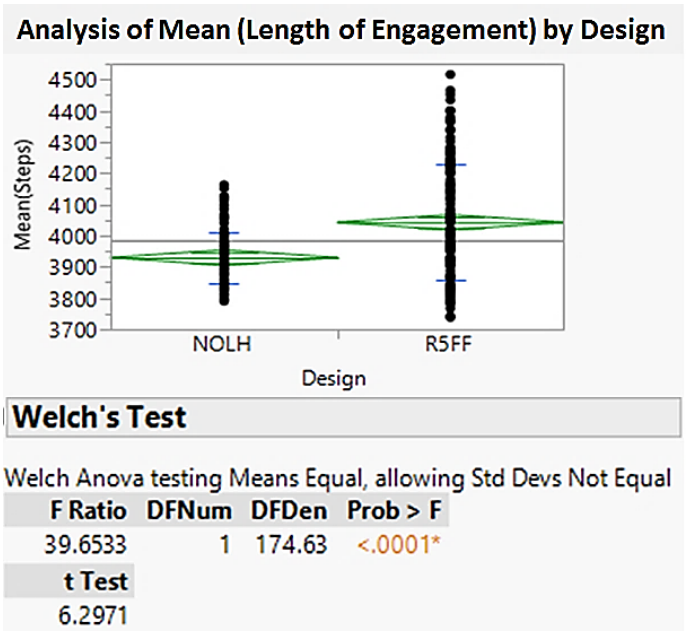


Figure 18. Analysis of Means (Length of Engagement) by Design

In Figures 16, 17, and 18, it can obviously be seen that there are numerical differences between the NOLH and R5FF output. These differences can be attributed to

the space-filling aspect of the NOLH design and the corner-point sampling of the R5FF design.

First, in order to compare the means of both data sets, we combined the summary data of the NOLH and the R5FF into one dataset. Then, we employed the Welch Analysis of Variance (ANOVA) test to determine if there was a statistically significant difference between the means of the two groups of data. By visual inspection we were able to determine that the variances of the two groups are not equal, which is the reason we utilize the Welch test instead of the usual t-test that assumes equal variances. The Welch statistic is based on the ANOVA F test. However, the means are weighted by the reciprocal of the group mean variances (Welch, 1951; Brown & Forsythe, 1974; Asiribo, Osebekwin, & Gurland, 1990). If there are only two levels, as is the case here, the Welch ANOVA is equivalent to the unequal variance version of the t-test. The null hypothesis for this test is that the means are equal. All of the resulting p-values are less than .05, so we reject the null hypothesis that group means are equal. We note, though, that the larger the number of design points (N), the greater the statistical power. And we must keep in mind that the greater the power, the greater the chance that even small, practically insignificant differences in the means are detected as statistically significant.

4. Linear Regression Analysis on Engagement Model Outputs (Block 1G of Work flow Diagram)

We performed a stepwise linear regression analysis on both the NOLH and R5FF MANA data sets. One goal was to better understand the experiment space by discovering the most influential parameters, their key threshold values, and potential interactions amongst them. This process also is useful for verification and validation of the model. In order to explore the most effective factors on red side's probability of win, we constructed metamodels for P(Win) of both engagement and campaign models. Another goal was to create metamodels for *a*, *b*, and engagement P(Win) , as functions of the Red and Blue experiment variables. As mentioned before, the metamodels for *a* and *b* will be plugged into campaign metamodel P(Win) = f(a,b), to yield a composite, embedded metamodel for campaign P(Win) = f(experiment variables).

a. Linear Regressions on the NOLH Data Library

First, attrition coefficients *a* and *b* are calculated from the number of casualties and length of engagement statistics, using the formulas in equations 2.7 and 2.8 given in Chapter II. Recall that *a* represents the killing power of *y* (Blue) on *x* (Red = "good guys"), and that *b* represents the killing power of *x* (Red) on *y* (Blue).

Stepwise linear regression models are fit to the estimates for *a* and *b*, to determine which factors best predict them. To fit the models, we utilized JMP's stepwise regression capability, and utilized the default minimum Bayesian information criterion (BIC) to determine when to stop adding terms to the regression model. We allowed all main effects, quadratic effects, and two-way interaction terms to potentially enter the model. Upon obtaining the final model, we utilize JMP's features to assess the assumptions of regression, namely that residuals are normally distributed with zero mean and constant variance.

We start with the regression model for *b*, shown in Figure 19.

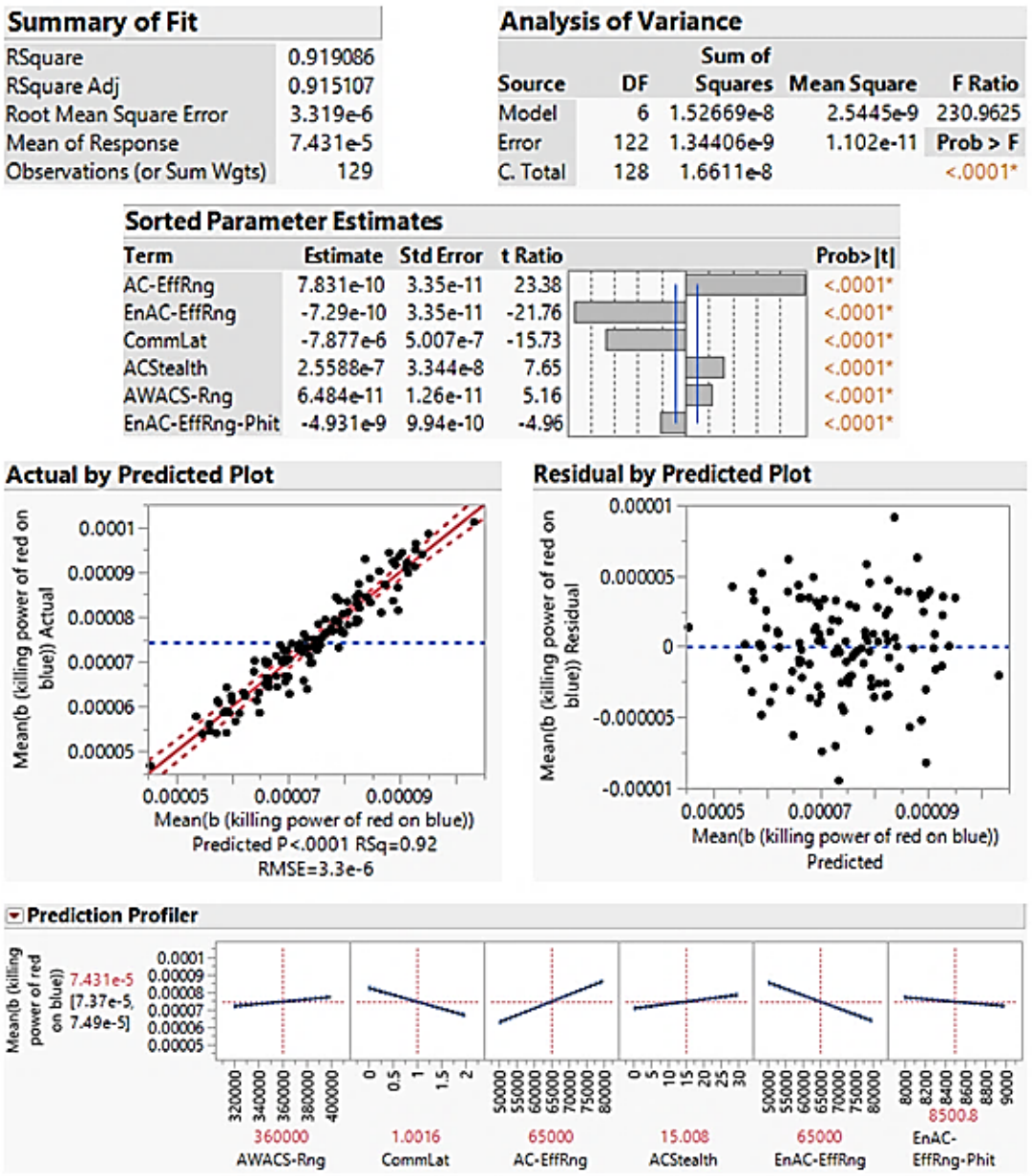


Figure 19. Regression Analysis of Red Killing Power on Blue (*b*) Based on NOLH Data library

We note that the RSquare value in Figure 19, is very high (0.919), which indicates that the model explains over 91% of the observed variability in the data. The Actual by Predicted plot additionally indicates that the model fits well. In the Sorted Parameter Estimates, the statistically significant variables and the size/direction of their effects on the response are shown. The Red aircraft's missile effective range is the most influential

driver of *b*, which confirms intuition. The Residual by Predicted plot indicates that the assumption of zero mean and constant variance is reasonable, since there is roughly even scatter above and below the zero line. Though we don't display it here, we also confirmed the normality of the residuals visually and through the goodness of fit test mentioned previously. The Prediction Profiler in Figure 19 illustrates the marginal effect of each predictor variable on the response, *b*. The absolute value of the slope of the line indicates the magnitude of the effect and the sign of the slope (positive or negative) indicates whether the effect has a positive or negative impact on the response as that variable is increased. JMP also allows us to save the prediction formula for *b*, and it is shown in Figure 20.

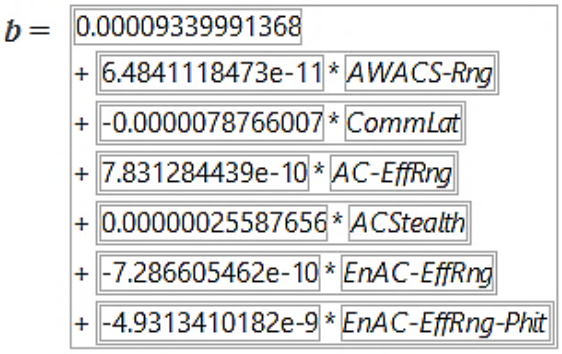


Figure 20. Prediction Formula for *b* Based on NOLH Data Library

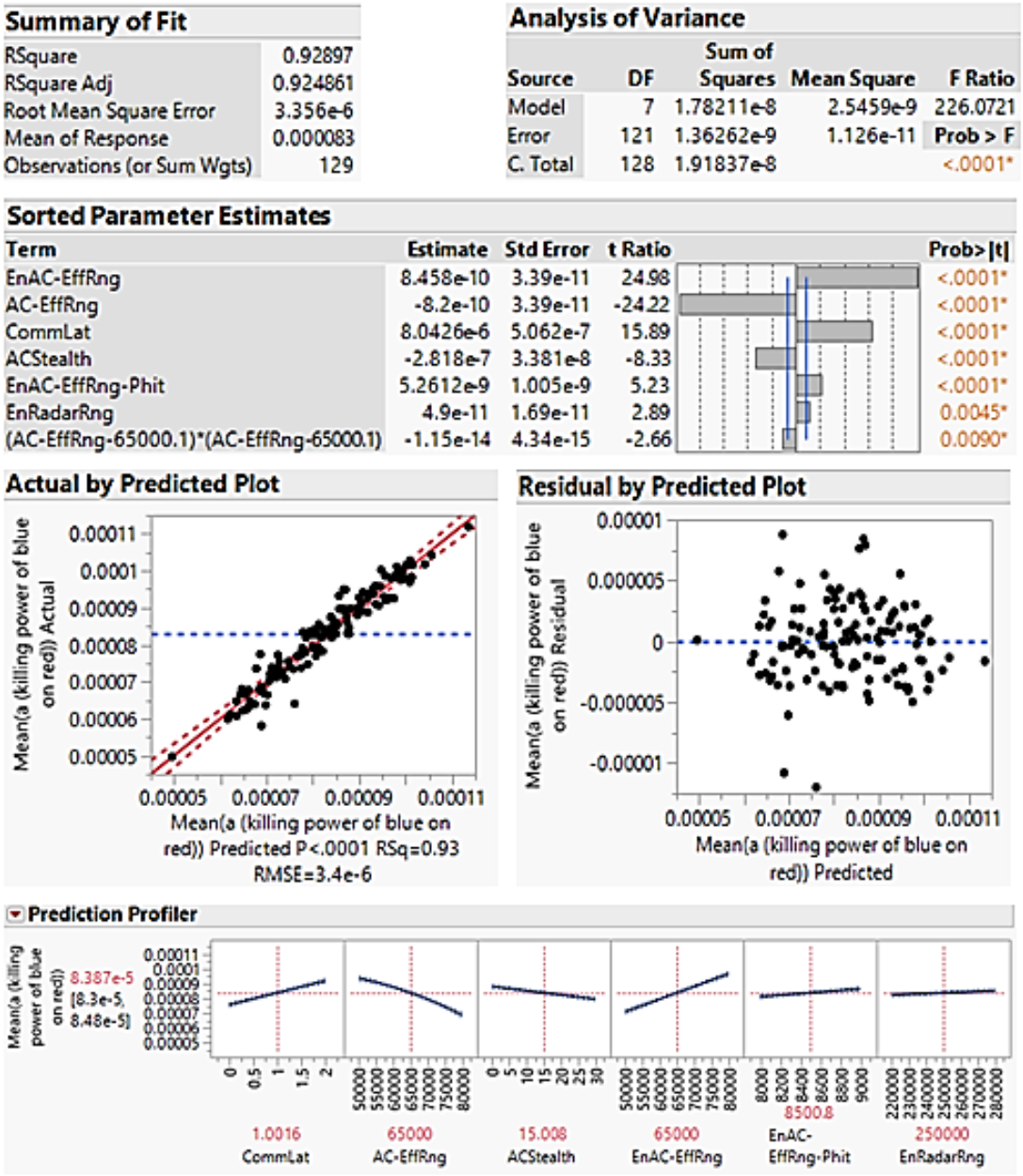


Figure 21. Regression Analysis of Blue Killing Power on Red (*a*) Based on NOLH Data Library

Next we fit a linear model for Blue killing power on Red, *a*. The resulting output is shown in Figure 21. Like the previous regression model, the RSquare value is also very high (0.928). In the Sorted Parameter Estimates, we see that the Blue aircraft effective range is most influential and has positive relationship with *a*, which confirms intuition.

The Prediction Profiler shows that there is a non-linear (quadratic) relationship between the Red aircraft effective range and the response, *a*. The diagnostic plots indicate that the model is well-fitting and that the assumptions of regression are upheld. The prediction formula for *a* is given in Figure 22.

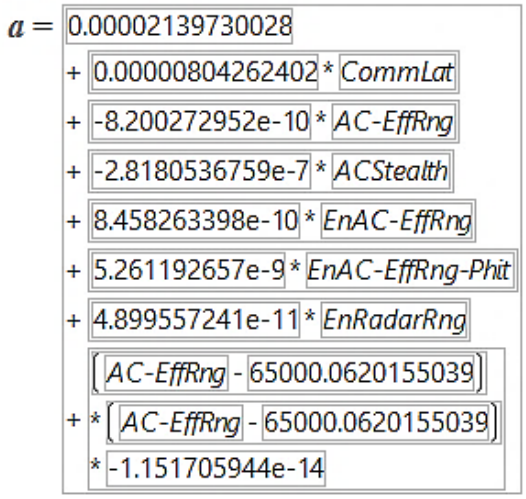


Figure 22. Prediction Formula for *a* Based on NOLH Data Library

Additionally, we fit a model for P(Win). As mentioned previously, P(Win) is the probability that our side, Red, wins—that is, experiences few casualties than Blue. Figure 23 shows the regression analysis statistics and plots for the final P(Win) metamodel. Here also, RSquare is high (0.926) and the model fits well. The most influential factors are the effective ranges of the aircrafts, and we note that the t-ratio of Red aircraft's effective range is slightly larger than the t-ratio of Blue aircraft's effective range.

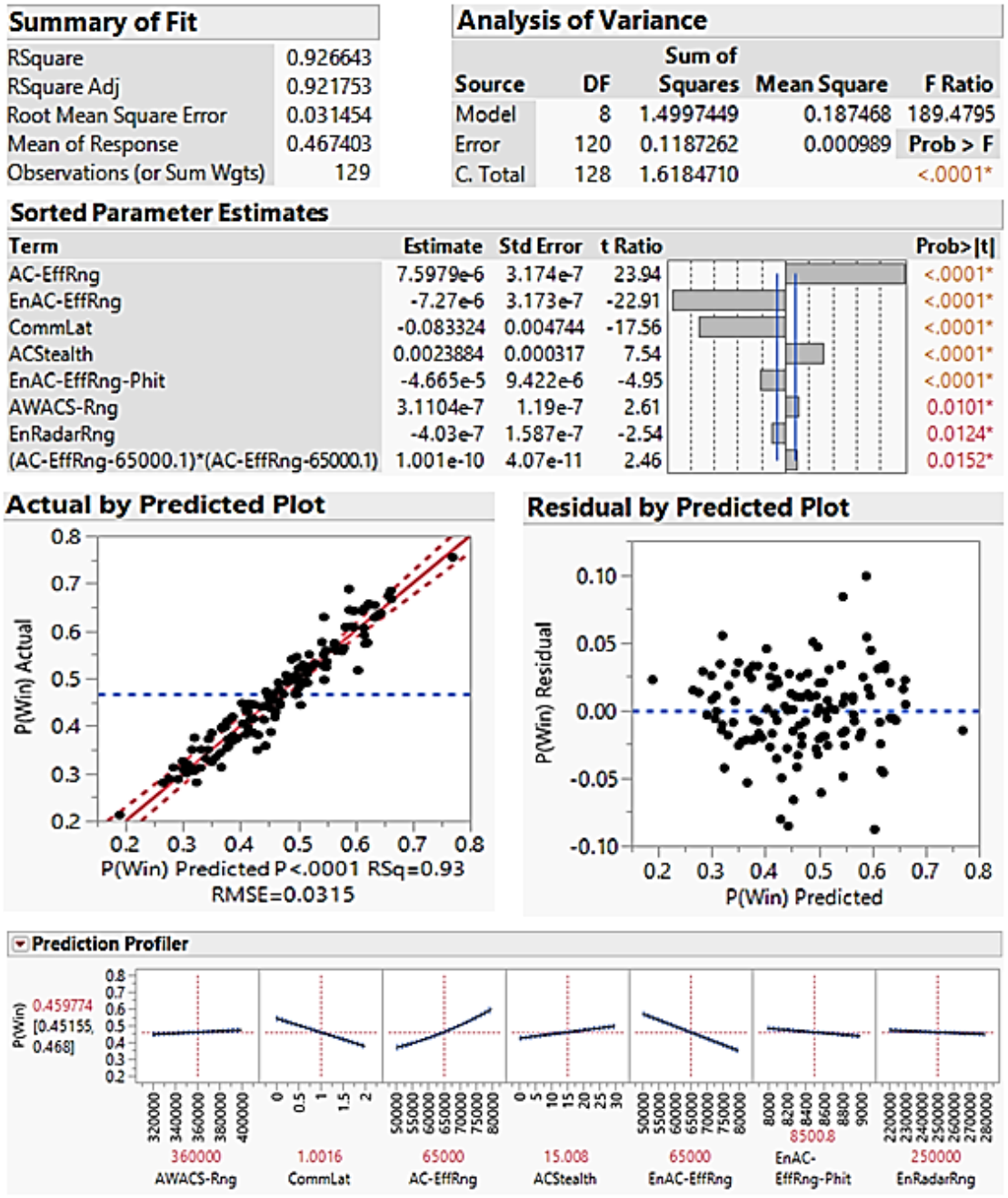


Figure 23. Regression Analysis of Engagement Model P(Win) Based on NOLH Data Library

As with the other regressions, this model fits well and the assumptions of regression are upheld. The direction and magnitude of the effects, illustrated in the Prediction Profiler, make sense. For example, Red engineering factors such as AWACS radar range, Red aircraft effective missile range, and Red aircraft stealth percentage all

have a positive relationship with P(Win), whereas Blue engineering factors and the communication latency of Red aircraft have a negative relationship with P(Win). It is also seen that the effect of Red aircraft effective missile range on P(Win) is quadratic. The prediction formula for P(Win) is given in Figure 24.

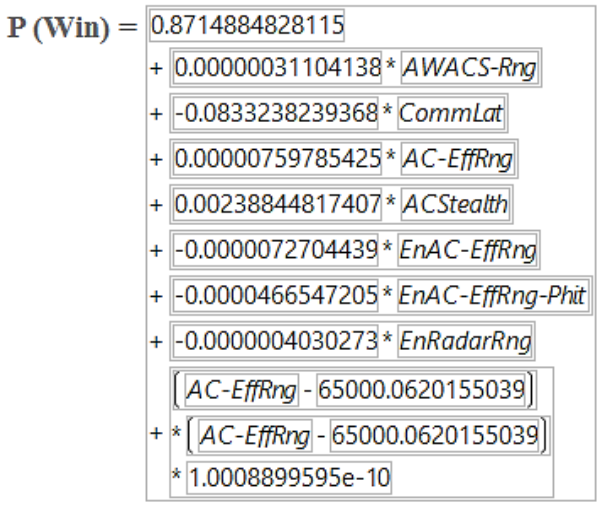


Figure 24. Prediction Formula for Engagement Model P(Win) Based on NOLH Data Library

b. Linear Regressions on the R5FF Data Library

The process of discovering metamodels, via stepwise regression, for *a*, *b*, and P(Win) are repeated, this time using the R5FF data library. The result of the final model for *a* is shown in Figure 25. The high RSquare value (0.982) indicates that our model fits well and explains most of the variability in the data. The diagnostic plots also indicate that assumptions are being upheld. The most influential driver of Blue's attrition power on Red (*a*) is the Blue aircraft's missile effective range. Red aircraft's missile effective range is a close second.

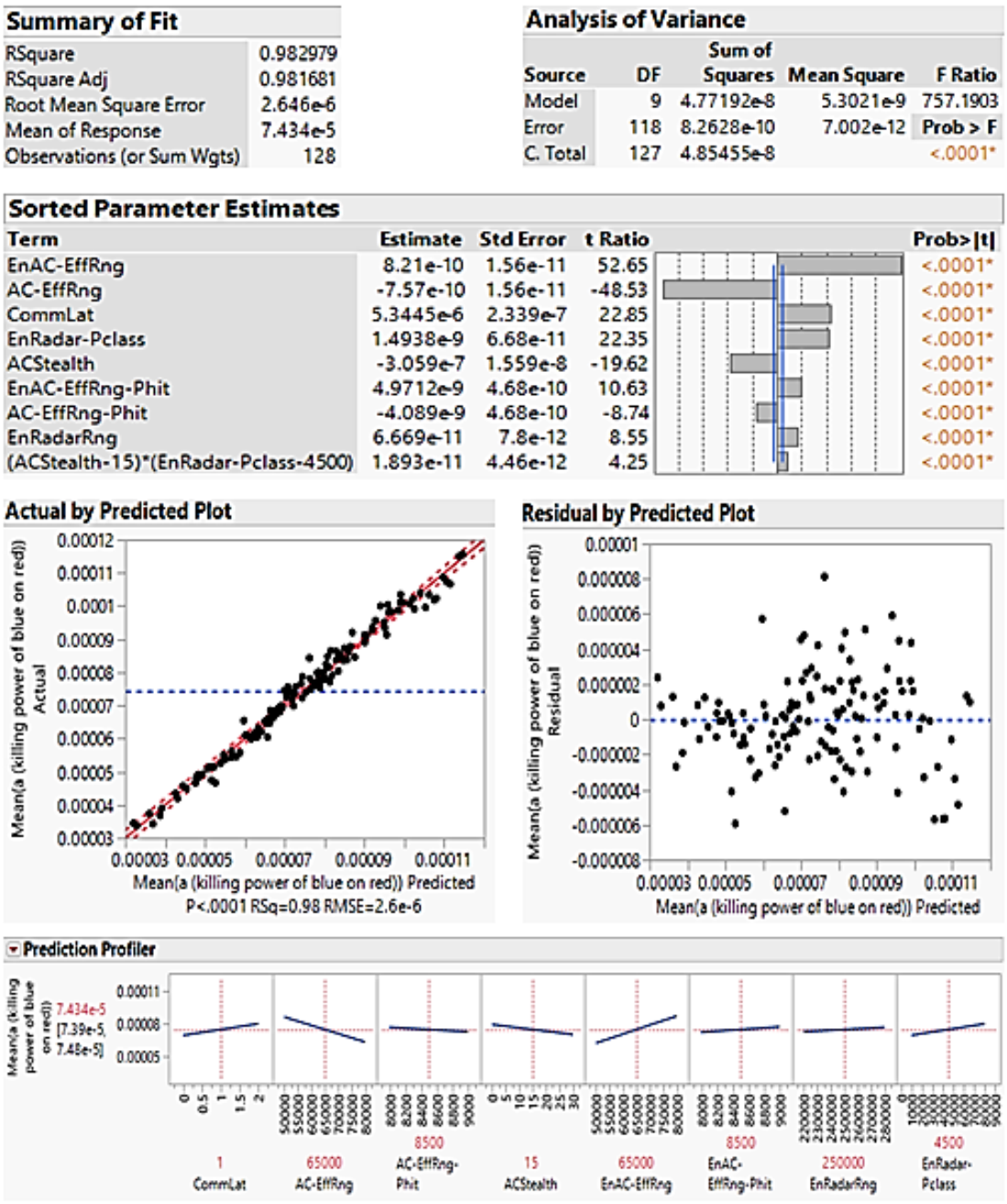


Figure 25. Regression Analysis of (*a*) Based on R5FF Data Library

The Prediction Profiler in Figure 25 shows very similar relationships to the ones that appear in the NOLH-based regression model. Though an interaction between ACStealth and EnRadar-PClass is statistically significant, and we leave it in the model for maximal predictive capability, the effect is not deemed to be practically significant.

Thus, we choose not to include the interaction plot in the results to conserve space. The prediction formula for *a* is shown in Figure 26.

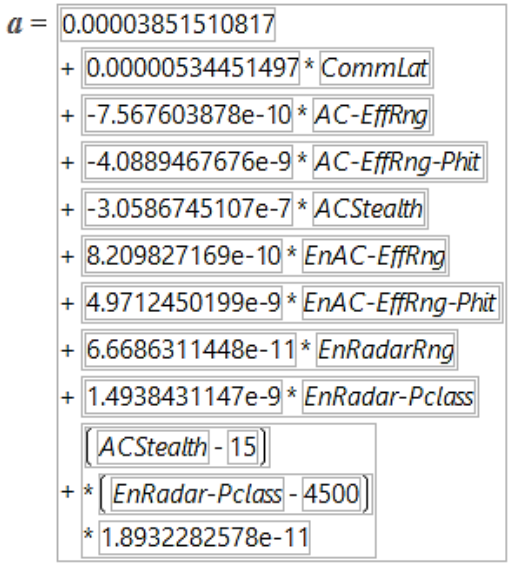


Figure 26. Prediction Formula for (*a*) Based on R5FF Data Library

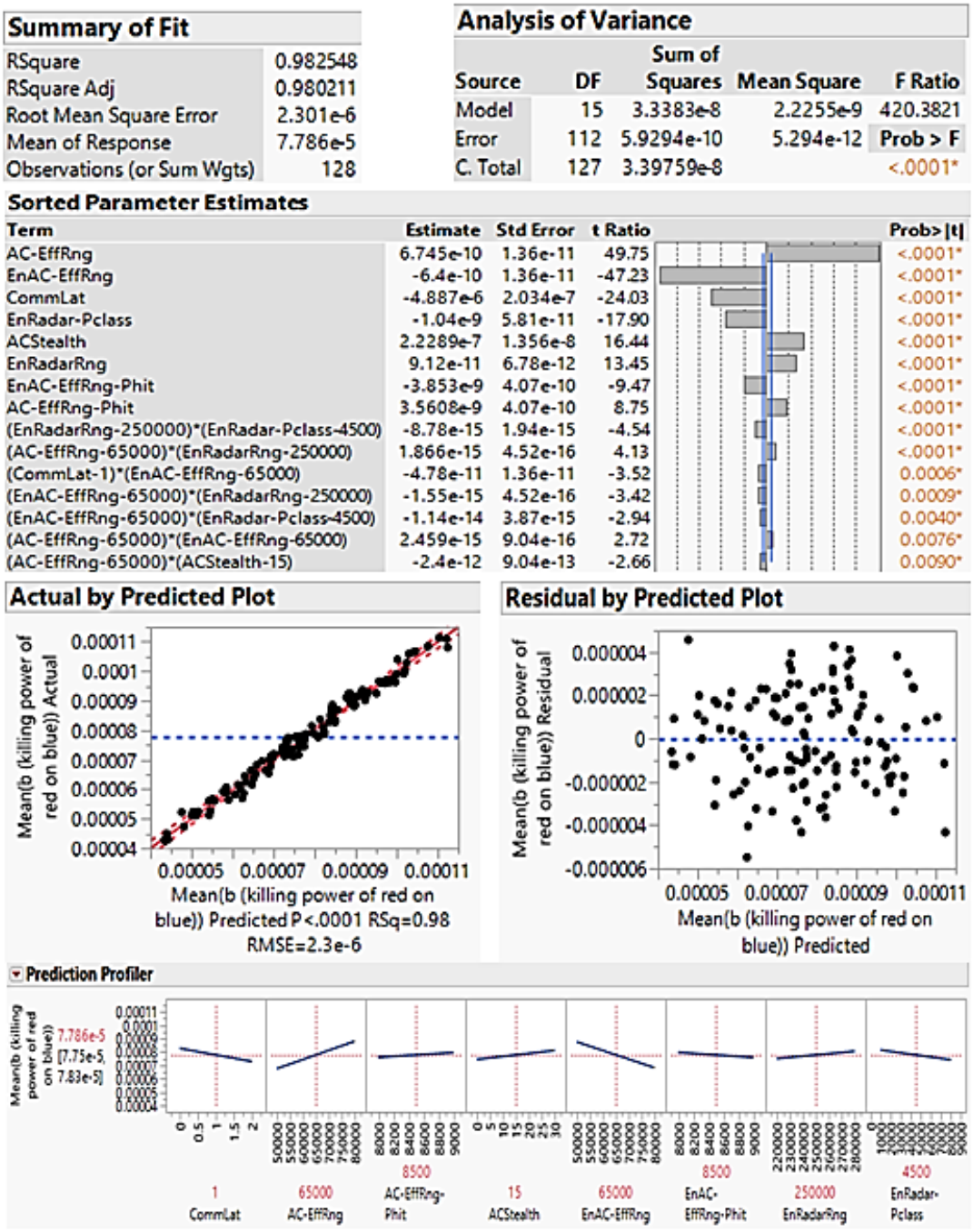


Figure 27. Regression Analysis of (*b*) Based on R5FF Data Library

Next, a regression model is fit for the killing power of Red on Blue (*b*), and the result is shown in Figure 27. The RSquare value (0.982) is, again, high, as in previous models.

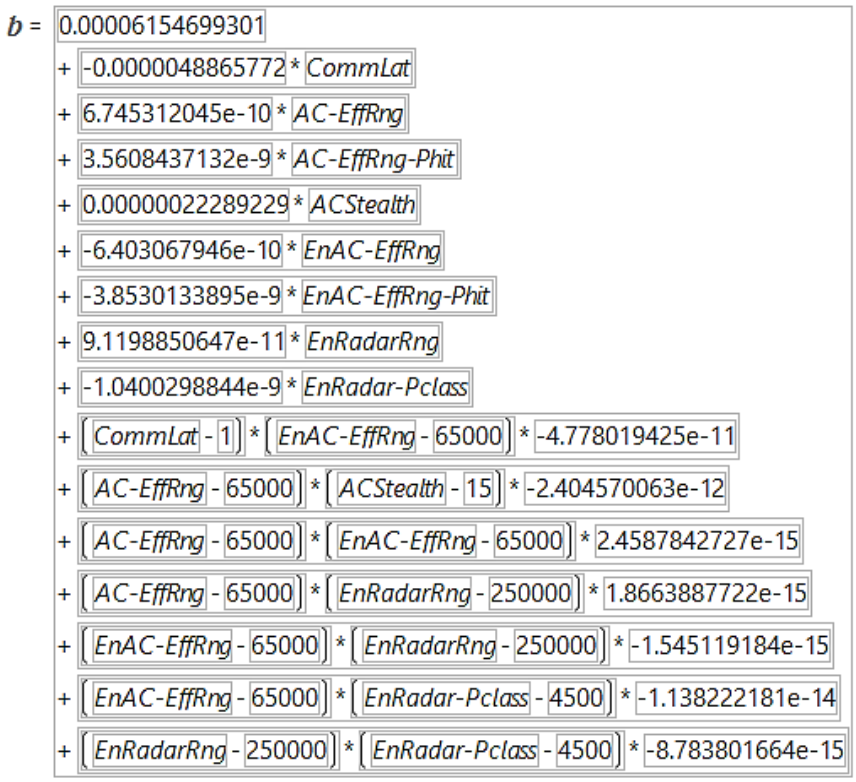


Figure 28. Prediction Formula for (*b*) Based on R5FF Data Library

Interestingly, this regression for *b* has more statistically significant terms than the equivalent model for *b* based on the NOLH data. Though not all terms are deemed to have practical significance, including a set of the interaction terms. As before, though, we are choosing to leave all statistically significant terms in the final model. The fact that more statistically significant terms were picked up with the R5FF as compared to the NOLH may be the result of the corner-sampling aspect of the R5FF, which tests parameters at intervals more extreme (thus, larger ranges) than the space-filling NOLH design. The diagnostic plots indicate that the assumptions of linear regression are satisfied. As seen previously, the effective missile range of both side's aircraft have the strongest effect on the response, *b*. The prediction formula for *b* is shown in Figure 28.

Finally, we fit a model for P(Win) based on R5FF data, and the result is shown in Figure 29. Again, the Rsquare value is very high (0.986), and the diagnostic plots look reasonable. The most influential factors are shown in Sorted Parameter Estimates.

Aircraft missile effective ranges have the largest impact on P(Win), which is reasonable and confirms intuition. The prediction formula is given in Figure 30.

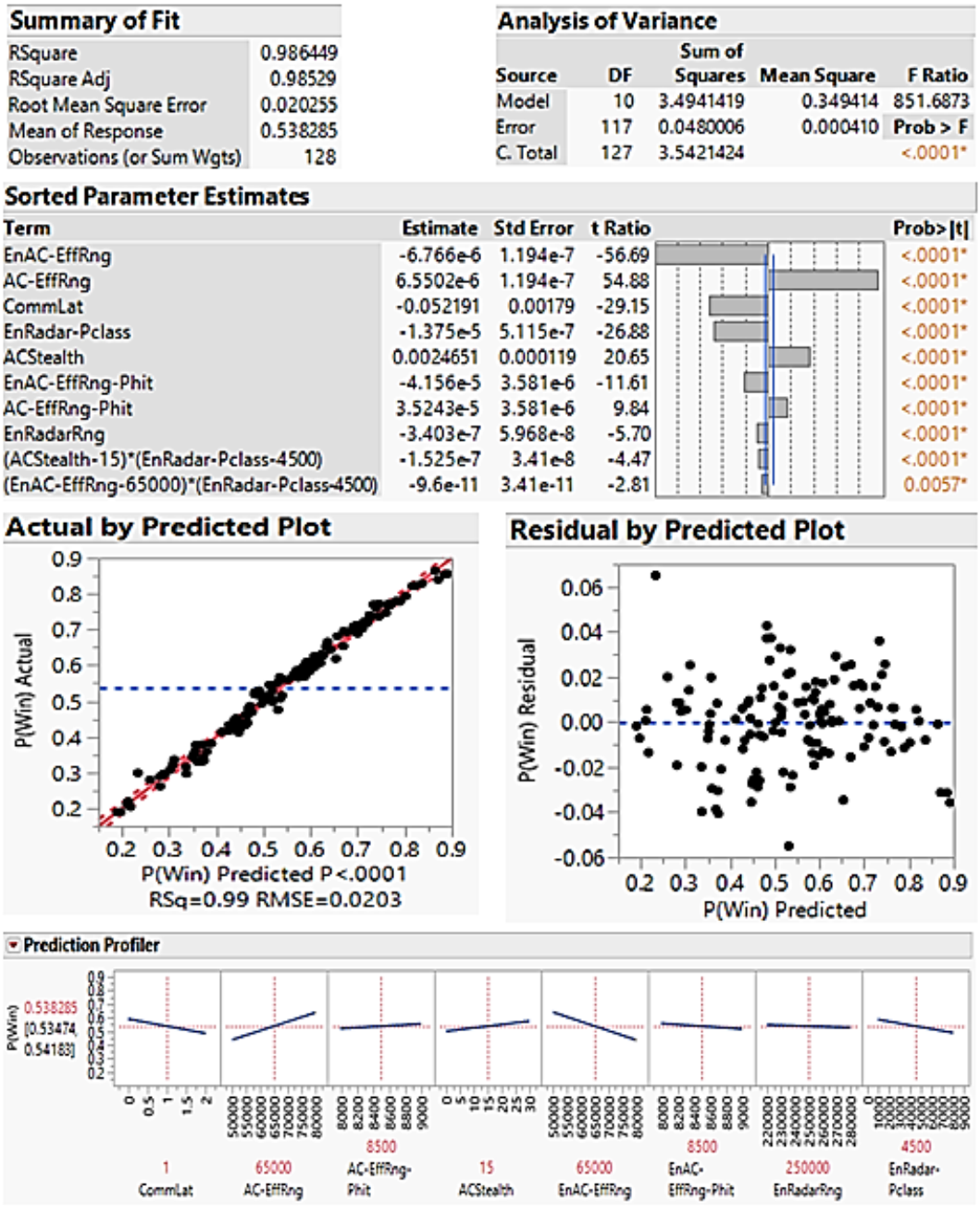


Figure 29. Regression Analysis of Engagement Model P(Win) Based on R5FF Data Library

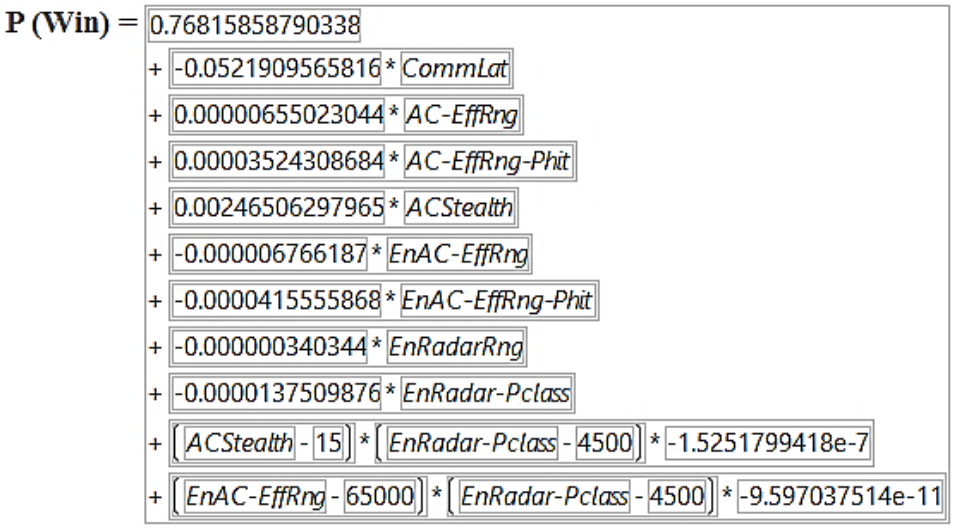


Figure 30. Prediction Formula for Engagement Model P(Win) Based on R5FF Data Library

c. Comparing the NOLH and R5FF Metamodels

When we compare the NOLH and R5FF metamodels, we conclude that they are largely similar in terms of the relative magnitude of the strongest effects, but that there are a few differences as well. As one example, the NOLH allowed us to detect the quadratic effect of the Red AC-EffRng on *b* and P(Win). The R5FF, sampling each factor at only two corner points, does not allow us to detect the quadratic effect. The R5FF, does, however, allow us to pick up a few other statistically significant effects, as the range of the parameters is generally larger than it is with the NOLH. In short, the choice of design technique provides different abilities to explore the design space. Time permitting, both designs can be added together to achieve both good space-filling and corner-sampling.

The prediction formulas for *a* and *b*, will be plugged into the metamodel P(Win) = f(*a*,*b*) that will be obtained from the result of running a designed experiment on the campaign model. This will yield a composite, embedded metamodel which will allow us to explore the effects of engineering-level decisions and changes on the outcome of campaign.

B. CAMPAIGN-LEVEL MODEL PROCESS

The campaign-level model is coded using Python Notebook, and implements the stochastic Lanchester Linear Law model presented in Chapter II. Metrics captured are Red Casualties, Blue Casualties, and P(Win). It simulates a 100-vs-100 aircraft campaign engagement between Blue and Red, with four discrete time periods. In each of the four time periods, 25 Red aircraft engage 25 Blue aircraft. Each side has a breakpoint for each time period. If a side experiences enough casualties as to reach their breakpoint, they will disengage the other force. The Python code reads in csv files of the NOLH and R5FF data libraries, and utilizes the different sampling methods shown in the work flow diagram (Figure 9) to perform replications of the model. Since we utilize different methods for sampling, there will be different output data files to analyze and compare. All the sampling methods and campaign simulations are done within the Python code. At the end of each campaign, the number of casualties is summed, and the side with fewer casualties is deemed the winner. For each stochastic replication of the campaign, the aforementioned campaign metrics are calculated and stored. The Python code is given in the Appendix.

1. Campaign Model Input Calculations and Input Analysis (Block 1F of Work Flow Diagram)

Utilizing the Lanchester linear law equations presented in section II.D.1 of this thesis, attrition coefficients for each force, (a) and (b), for each run are calculated using formulas that we now re-present below for convenience in equations 4.1 and 4.2. Also these equations are utilized for all campaign models of each sampling method.

$$a = \frac{x\ casualties}{(time)\cdot(one\ x\ participant)\cdot(one\ y\ participant)} \tag{4.1}$$

$$b = \frac{y\ casualties}{(time)\cdot(one\ x\ participant)\cdot(one\ y\ participant)} \tag{4.2}$$

The work flow diagram contains seven different sampling and DOE methods applied to the campaign model. Though we discuss the details of these seven methods in the remainder of this chapter, here we show in Figures 31 and Figure 32 the side by side boxplots of the a and b values for these seven methods. Since we have not yet presented

all of the details about these individual methods, the main take-away at this point is simply to notice how different the input ranges are across these seven methods. Therefore, we should expect large and practical impacts on the campaign results.

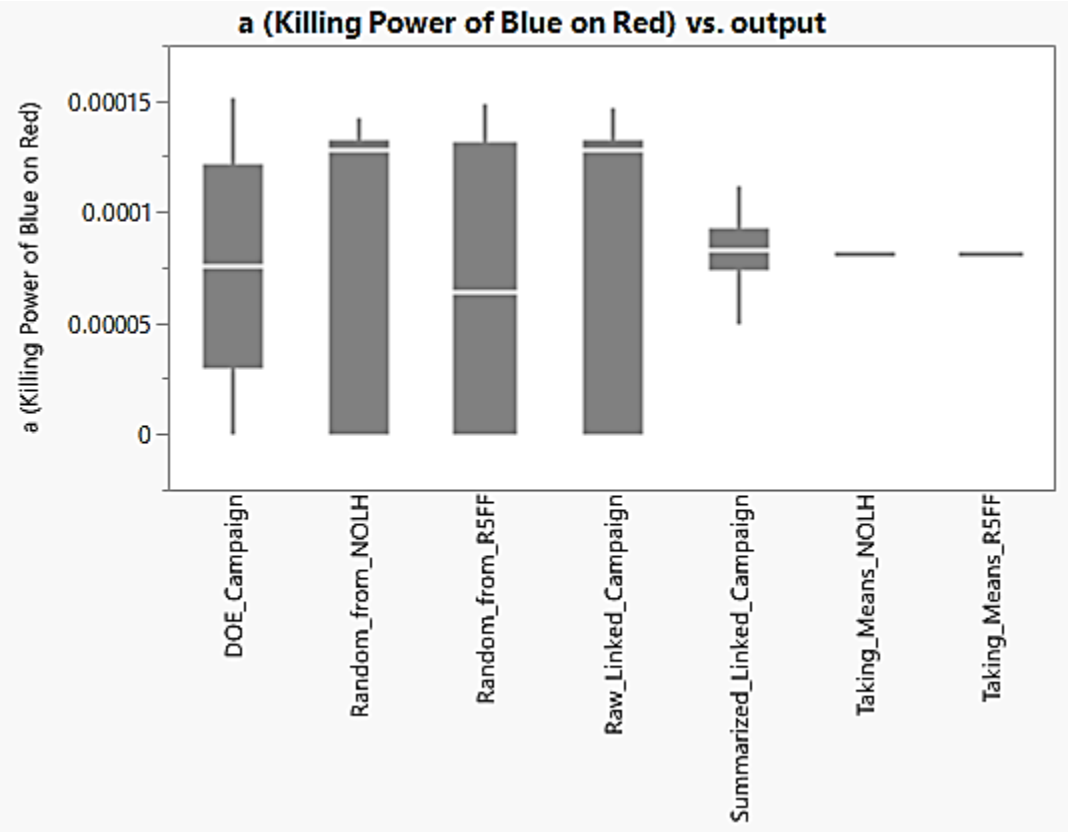


Figure 31. Comparison of (*a*) Value Obtained from Different Sampling Methods and Experimental Design

Figure 31 shows the difference among *a* inputs obtained by employing seven different methods. We note that the "Summarized_Linked_Campaign" and the two "Taking Means" results have similar medians. We can also see that "Random_from_NOLH" and "Raw_Linked_Campaign" have skewed distributions.

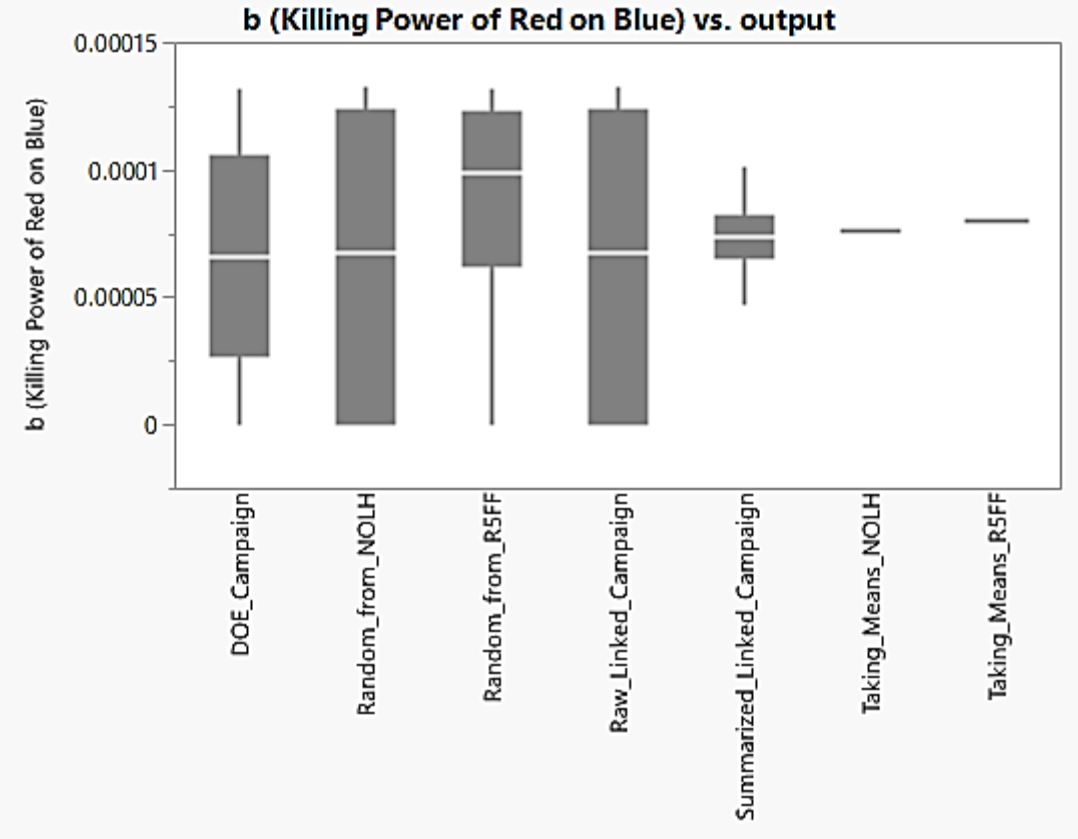


Figure 32. Comparison of (*b*) Value Obtained from Different Sampling Methods and Experimental Design

In Figure 32, we show the box plots of *b* inputs for the seven methods. Except for the skewed output from the "Random_from_R5FF" method, we note that the other methods produce numerically similar medians.

2. Campaign Model Runs by Employing Sampling Methods and Output Analysis (Blocks 2A, 2B, 2C, and 2D of Work Flow Diagram)

A function in python, called "lanch" is written, which simulates one of the four engagements (time periods) in an air-to-air campaign. This function is therefore called four times to simulate an entire air campaign. One engagement consists of 25-vs-25 aircraft, fighting until one side reaches its breakpoint, which is six remaining aircraft, a quarter of the operation's force size. For every method of running the campaign model, according to the work flow diagram, the lanch function is called, the specified number of stochastic replications are performed, and the campaign outputs are written to csv files. It

takes approximately four seconds to run the campaign model, with the author's laptop having the features listed in section II.E.4.

a. Using Overall Sample Mean of MANA Outputs (Block 2A of Work Flow Diagram)

First, the campaign model is run for 10,000 replications, using a single point estimate for *a* and a single point estimate for *b*. The point estimates are generated using the overall sample mean (over the MANA library) of Red casualties, Blue casualties, and length of the engagement within equations 4.1 and 4.2.

Since we use point estimation by using the overall mean of the MANA model outputs, running the campaign model merely provides us with a single point estimate of P(Win), but as well a data library (with 10,000 rows) of Blue and Red casualties. With 10,000 replications, the standard error associated with P(Win) will be no greater than .005.

We run the campaign model over 10,000 replications two times, once using the sample means obtained from the NOLH data library, and once using the sample means from the R5FF data library.

(1) Taking Overall Sample Mean Input from NOLH Data Library

Running the campaign model with overall sample means from the NOLH data library provides the histogram and summary statistics shown in Figure 33. We note again that we obtain a single point estimate (value) for P(Win). Our estimate for P(Win) is .341. This point estimate would of course change if the model is re-run for another set of stochastic replications. Having only one point estimate is not a reasonable and plausible outcome for an analyst whose goal is to provide insight to a decision maker which takes into account uncertainty in the outcome of the campaign. On the other hand, this sampling technique does provide distributions, vice a single point estimate, for Red and Blue casualties. Having an entire distribution of data does provide an adequate means to understand uncertainty, and thus, risk. As seen in Figure 33, the two output histograms (Red casualties and Blue casualties), in this case, are left-skewed. And we also notice that Red takes more casualties than Blue, on average. The "spike" that occurs at 76 Red

casualties, this is the case where Blue wins every battle. A similar but smaller spike can be seen in the Blue casualties histogram.

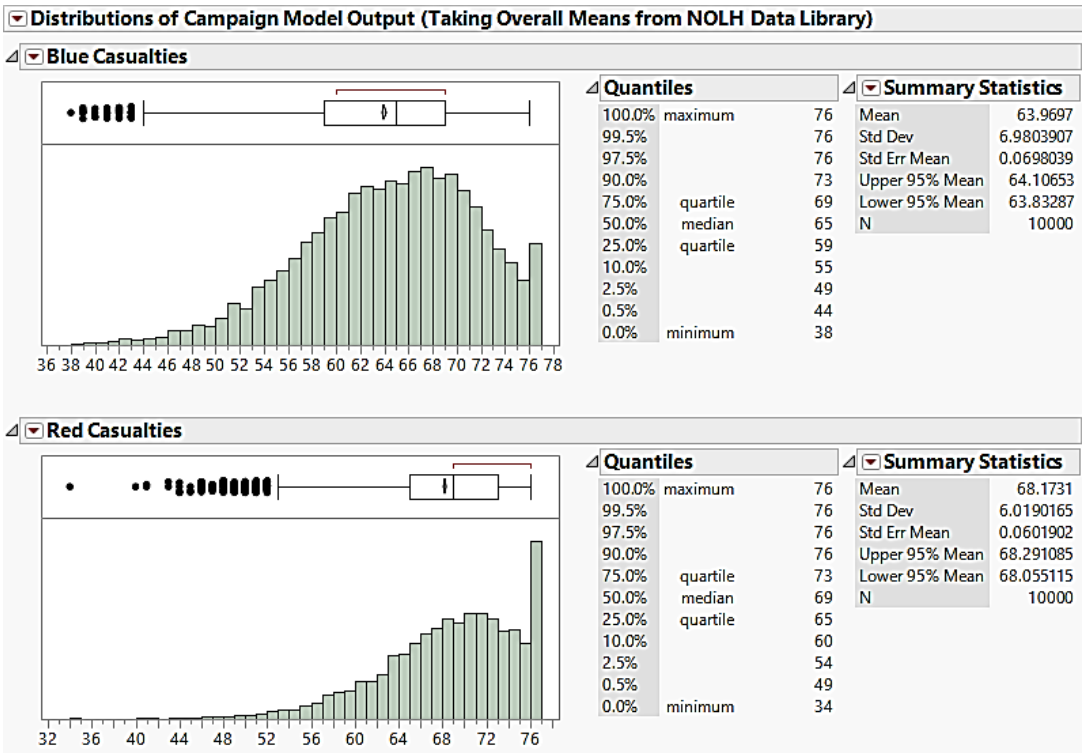


Figure 33. Distributions of Campaign Model Output (Taking Overall Sample Means from NOLH Data Library)

(2) Taking Overall Sample Mean Input from R5FF Data Library

Next, we used the overall sample means of Red and Blue casualties as well as length of engagement from the R5FF data library. The resulting histograms and summary statistics are shown in Figure 34. As before, instead of a distribution, we obtain a single point estimate for P(Win), in this case equal to 0.461. And, we again note that this point estimate would be expected to change if the model is re-run.

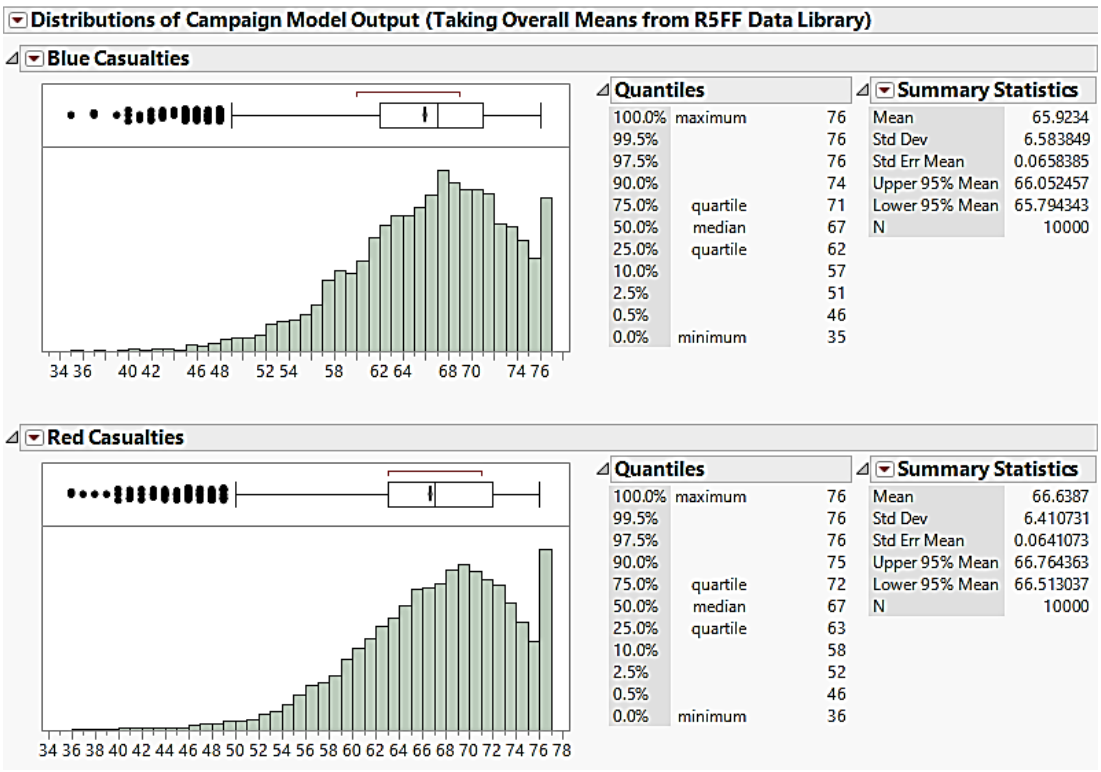


Figure 34. Distributions of Campaign Model Output (Taking Overall Sample Means from R5FF Data Library)

In Figure 34, we observe distributions that are similar to those generated using the NOLH data library. The summary statistics are also similar to those from the NOLH data.

When we compare histograms and summary statistics of the output from both designs, we observe that there are differences, such as differences in the sample means for Red and Blue casualties and slight differences in the shapes of the distributions. We can attribute these differences to the space-filling vs. corner-sampling aspects of the NOLH and R5FF designs, respectively.

More important than the slight differences between the two design outputs, however, is the main point that simply generating a single point estimate for P(Win) is insufficient, in terms of being able to properly evaluate uncertainty and risk.

b. Random Sampling (Block 2B of Work Flow Diagram)

In the previous step, we generated single point estimates for *a* and *b*, based only on sample means from the MANA data. In contrast, in this step, we utilize random sampling to provide a range of *a* and *b* estimates as input to the campaign model. We will do 1,000 different stochastic Lanchester "runs," where on each run, we will randomly select a different pair of *a* and *b* values from the MANA output for each of the 4 engagements in the campaign model. In other words, in total, we will randomly sample 4,000 rows (pairs of *a* and *b* values) from either the NOLH raw output or the R5FF raw output. Each run utilizes four of these pairs, one for each engagement time period. Likewise, the lanch function is utilized for each 25-vs-25 engagement. The sampling is done uniformly, without replacement. Recall that the NOLH raw output has 129,000 rows that could potentially be selected and the R5FF raw output has 128,000 rows that could potentially be selected. For each run, we perform 100 stochastic replications.

(1) Random sampling from the NOLH Data Library

Here we select pairs of *a* and *b* values from the NOLH raw output data library. The resulting histograms and summary statistics, for P(Win), Blue casualties, and Red casualties, are shown in Figure 35.

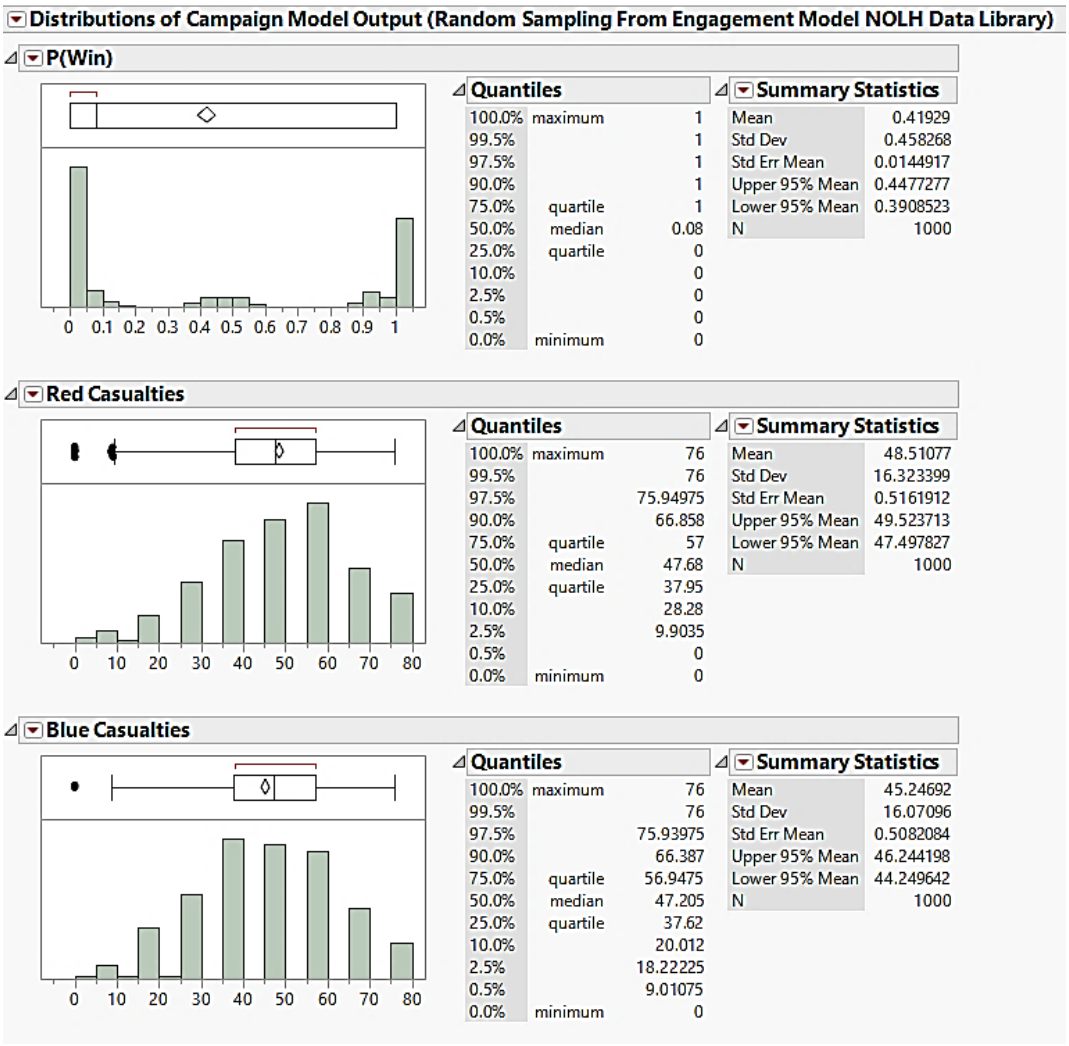


Figure 35. Distributions of Campaign Model Output (Random Sampling from MANA NOLH Raw Data Library)

We can make some interesting observations about the histograms in Figure 35. First, most strikingly, P(Win) is bimodal. We note how misleading it would be to interpret the point estimate of P(Win)=0.341, generated with the previous method, as "likely" or "representative," when obviously that point estimate is not at all representative of the set of possible outcomes that could occur. In other words, the risk of making decisions based on a single point estimate is very high. What would be extremely valuable to the decision maker, besides understanding the distribution/range of outcomes as we do here, is to conduct designed experimentation in order to better understand which

factors are driving success/failure. That is what we accomplished via Steps 1A through G of our work flow.

We also note that there are some small gaps in the histograms, which may be an artifact of the random manner in which the *a* and *b* values were sampled.

(2) Random sampling from the R5FF Data Library

The same random sampling process is next performed on the R5FF data library, in order to provide input to the campaign model. The result is shown in Figure 36. It can be seen that, again, there are gaps between bars in the casualty histograms. An interesting difference, as compared to the NOLH random sampling, is that the average P(Win) is substantially higher than it was with the R5FF data. This is an interesting outcome that shows how the choice of design can affect the results obtained. We once again advocate appending an R5FF design to an NOLH design, if time permits. We should also keep in mind that in this method we have only sampled a small portion of each of the raw data output libraries, that is, 4,000 samples taken from 129,000 rows for the NOLH and 128,000 rows for the R5FF.

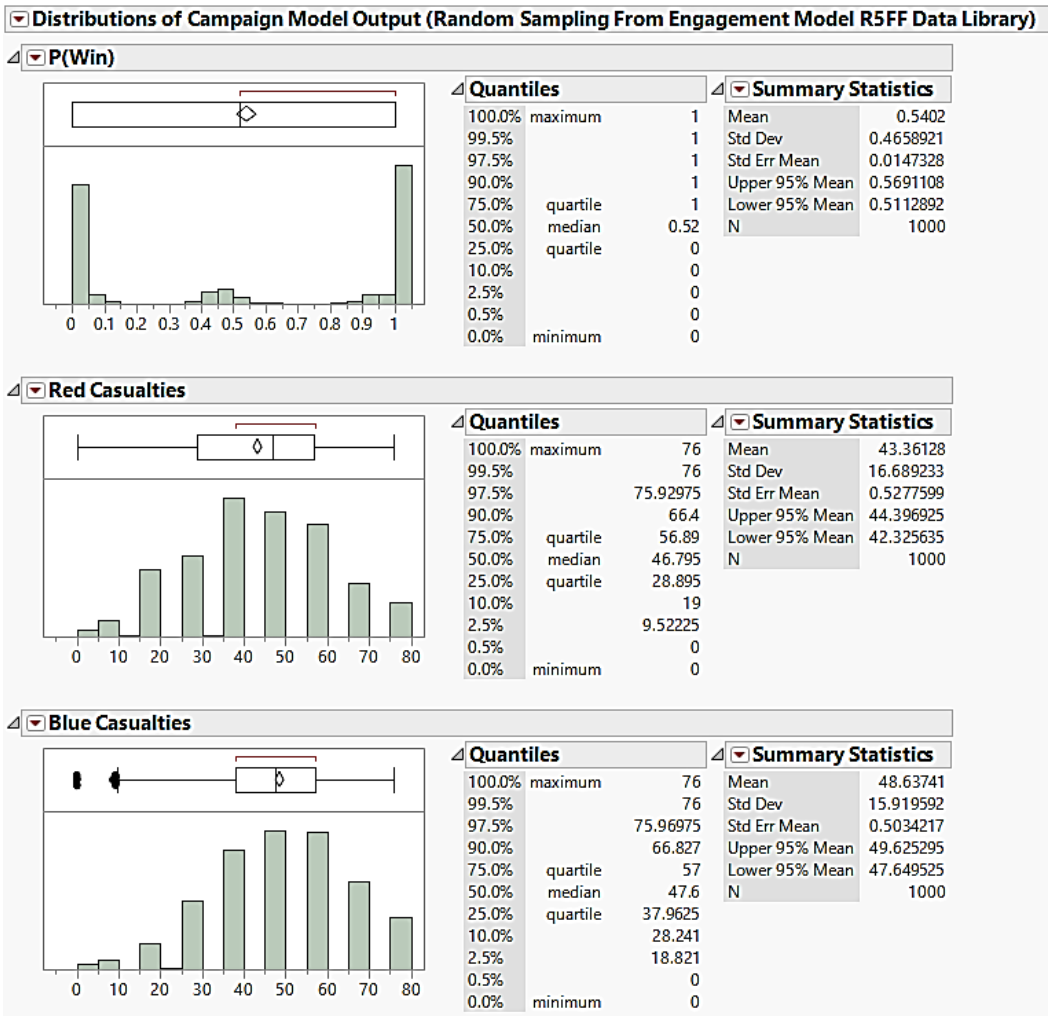


Figure 36. Distributions of Campaign Model Output (Random Sampling from MANA NOLH Data Library)

Similar interesting observations are observed in the histograms in Figure 36. As in previous random output, P(Win) has bimodal distribution. Here we again can say that applying point estimation to random MANA data output will be underestimating the uncertainty and risk. Understanding the distribution of outcomes is important, however, transferring the features of the distributions to higher level model is more important. The gaps in the histograms can be seen in Figure 36.

c. Linking Engagement and Campaign Models Directly (Linked Metamodeling; Block 2C of Work Flow Diagram)

We now describe Block 2C of the Work Flow Diagram, where we simulate the "direct linking" of the MANA engagement level output to the campaign model. We will demonstrate this method using the NOLH data, in order to take advantage of the space-filling feature of the design. Linking will work as follows: over the entire NOLH design space, run the Stochastic Lanchester model for every pair of *a* and *b* estimates (calculated from the MANA data), performing 30 random replicates for each pair. The lanch function is utilized for each period of campaign. This is in contrast to the previously described random sampling method, in that we will not sample a small portion of the NOLH space, but the entire space. We will perform this method using both the raw (full) data output library as well as the summarized data library. This method is analogous to running the campaign model with the ability to "call out to MANA" for a set of replications in the engagement model for some specific setting of design factors. This method would in actuality require software to link the two models and allow for "on demand" requests for MANA runs. Instead, what we have done here for simplicity is pre-generate the library of MANA data across the entire design space of interest.

(1) Linking Raw (Full) NOLH Data Library to Campaign Model

For each of the 129,000 calculated *a* and *b* values which are obtained from the engagement model raw NOLH data library, we performed 30 replications of the stochastic Lanchester model. Same *a* and *b* values are used for all periods of the campaign model. As a result, we obtain 129,000 rows of output, which include the means of our campaign metrics over the 30 replications.

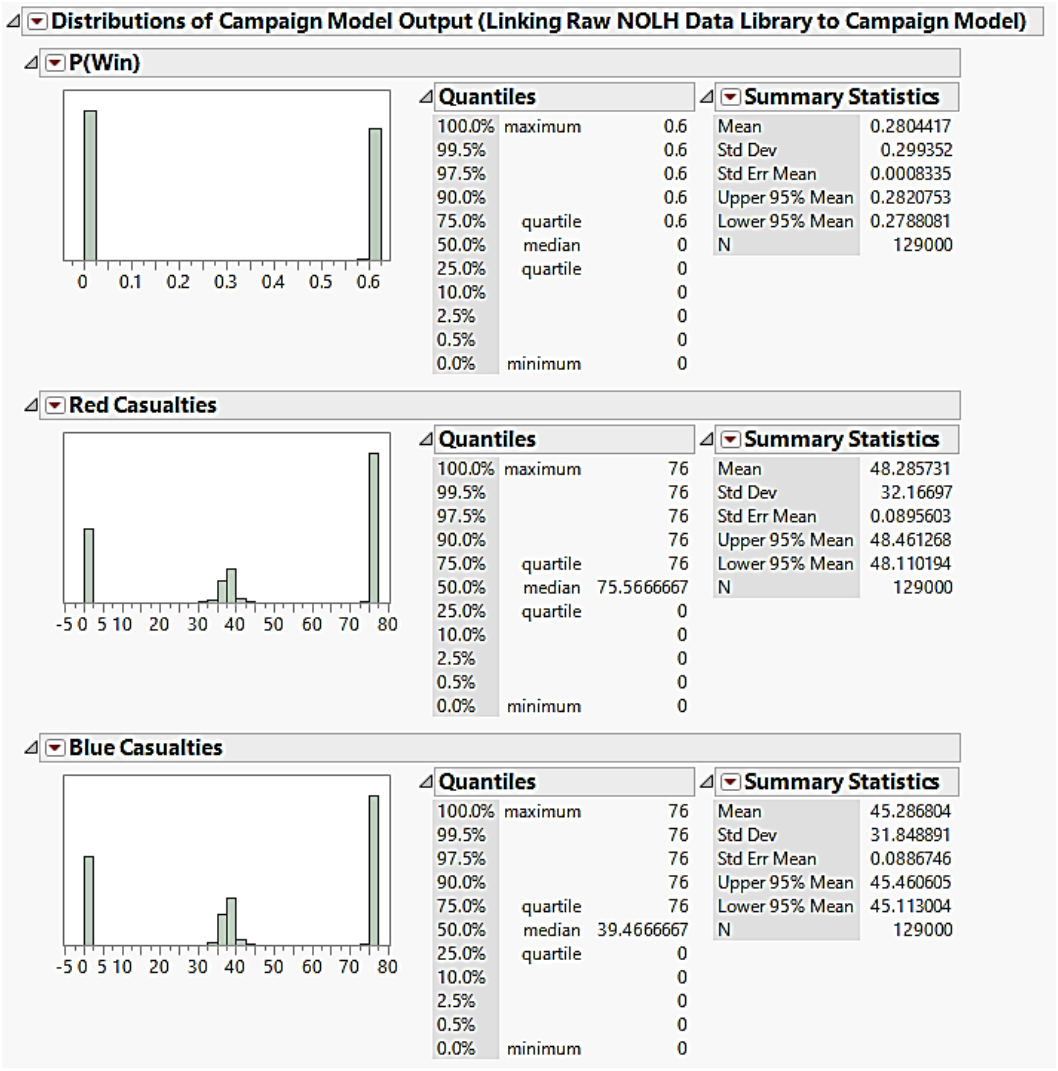


Figure 37. Distributions of Campaign Model Output (Linking Raw (Full) NOLH Data Library)

In Figure 37, the P(Win) histogram shows how the probability of win "piles up" on only two outcomes, 0 and 0.6. At first, this seems like an unrealistic outcome. However, when we analyze the data we understood that this feature of the P(Win) histogram is the artifact of MANA model choice. Since we decided two have a 2-versus-2 engagement, there are only two outcomes of this engagement for each side. One side can reach its goal (returning to base) if the other side's fighters are shot down. In other words, one side wins either by not losing any fighters or losing one fighter when the other side loses both of its fighters. Therefore, the number of casualties can be only 1 or 2 for

each side. As mentioned before, we calculate the attrition coefficients for the campaign model, *a* and *b*, using this number of casualties. Having a lack of variability in the casualty numbers impacts the calculation of the attrition coefficient values. As a result, one side has superiority and therefore every time wins or loses campaign, which is why there are only two bars in the P(Win) histogram. Summary statistics are not sufficient in this case since the mean is not representative of the distribution of outcomes, and the result is very different from the output of other sampling techniques.

So, even though this method utilizes the entire data library, it does not provide output to reasonably assess risk and uncertainty. We note that in order to increase variability in P(Win) and number of casualties, the number of fighters in the engagement model could be increased beyond two per side, so that there would potentially be more combinations of Blue and Red casualties observed in the data.

(2) Linking Summarized NOLH Data Library to Campaign Model

Secondly, we run the stochastic Lanchester model 30 times for each of the 129 *a* and *b* estimates which are obtained from the summarized (by means) NOLH data library. Likewise, same *a* and *b* values are utilized for all periods of the campaign model. As a result, we obtain 129 outputs which include the means of our campaign metrics over the 30 replications. The results are shown in Figure 38.

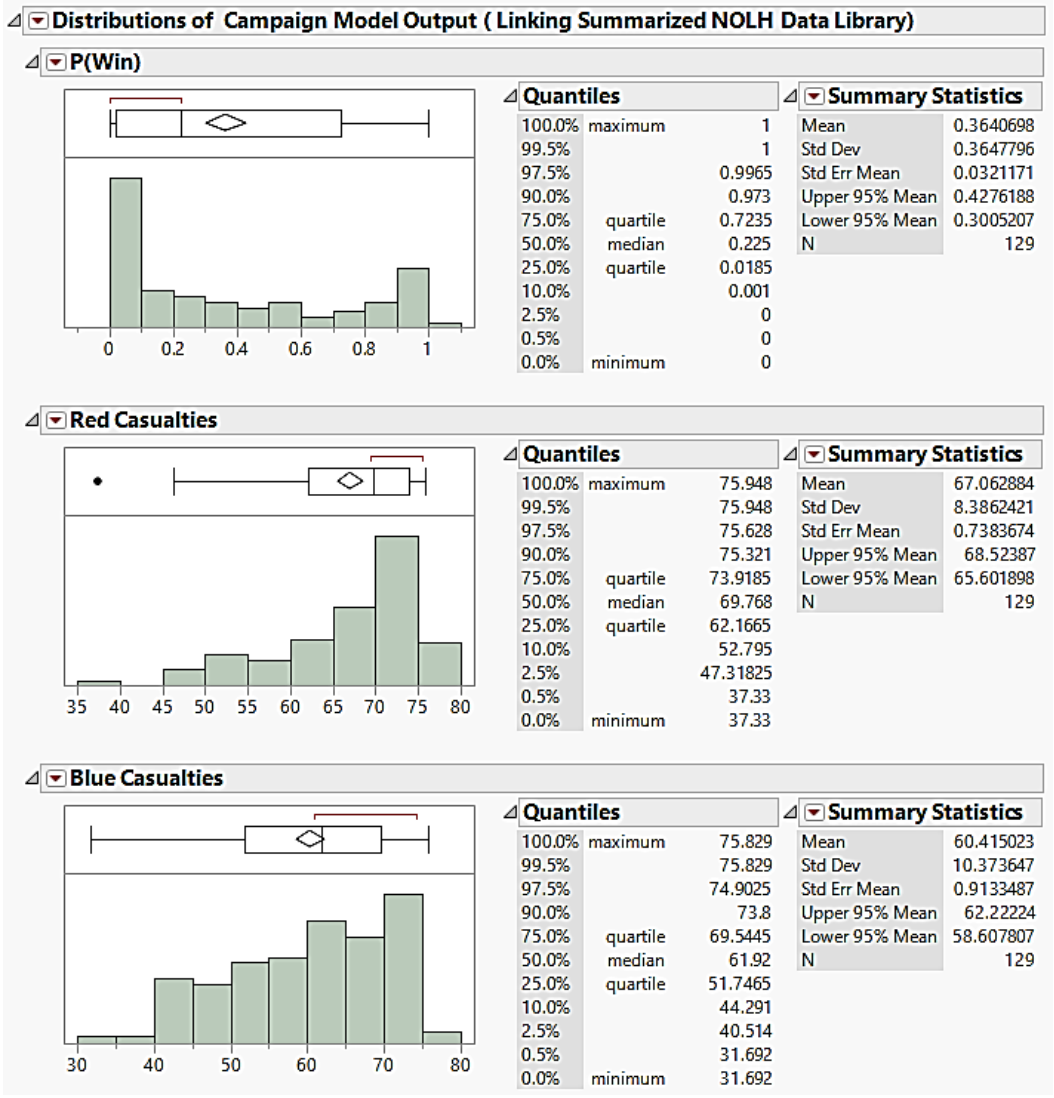


Figure 38. Distributions of Campaign Model Output (Linking Summarized NOLH Data Library)

The distribution histogram of P(Win) in Figure 38 now illustrates much more variability in the campaign outcome than in the previous step, which of course provides us a better assessment of uncertainty. The average value of P(Win) is 0.364, so Red has low probability of winning. Also, the spike on zero in the P(Win) histogram shows that Red has no chance of winning for a substantial number of runs (i.e., design points). In the summary statistics for P(Win), it is seen that the standard deviation of P(Win) is 0.364 which is very close to the mean, 0.364. This can be interpreted that there is large variability in P(Win) compared to the mean. Additionally, the mean of Red casualties

(67.068) is greater than the mean of Blue casualties (60.415), however, Red casualties has a bit lower standard deviation (8.386) than Blue casualties (10.373). Histograms of casualties for both sides are left-skewed. The spike on the 70–76 interval in the casualty histograms shows that one side loses most of its aircrafts in the campaign, which is an expected outcome when a Lanchester linear law is applied with break points. The variability in casualty histograms represents the risk in the campaign.

d. Constructing a Composite (Embedded) Metamodel for P(Win) (Blocks 2D.1, 2D.2, and 2D.3 of Work Flow Diagram)

(1) Performing a DOE, varying *a* and *b*

In this step, we conduct an experiment to vary two inputs to the campaign model, *a* and *b*. Note that we choose to fix the number of aircraft on each side. We use the ranges of *a* and *b* from the R5FF raw data library to generate the ranges over which *a* and *b* will be varied in the experiment. The range of *a* or *b* is simply its maximum value minus its minimum value. The reason we use the R5FF raw data, instead of the NOLH raw data, is that the R5FF tests the corner points while the NOLH tests the interior of the design space. We wanted to have the maximum possible (while remaining realistic for our model) ranges for *a* and *b*, therefore, we determined the use of the R5FF data to be more appropriate since it tests at more extreme points than the NOLH.

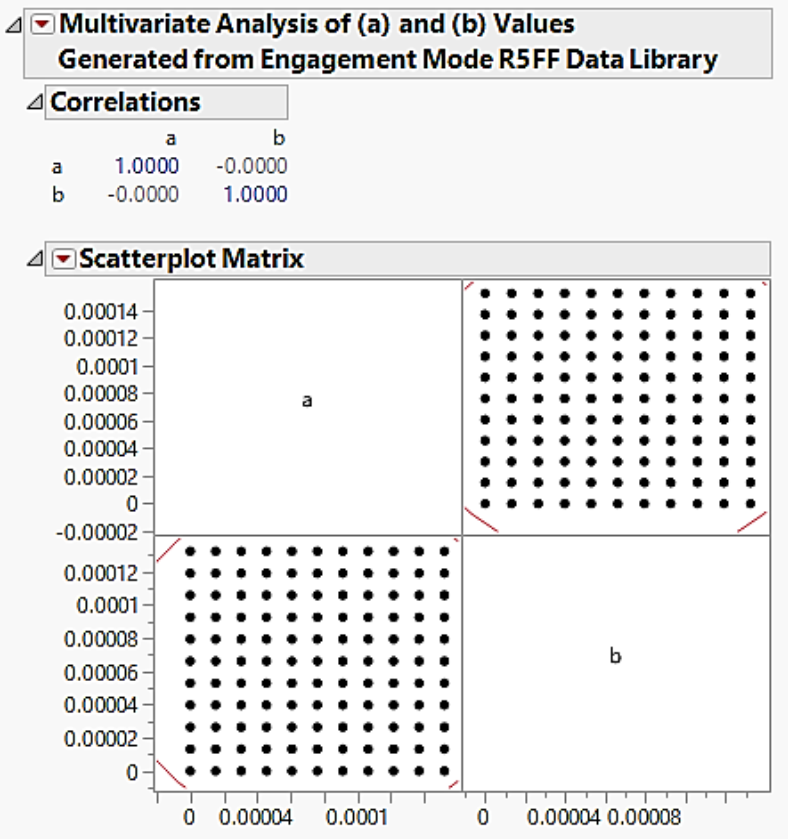


Figure 39. Multivariate Analysis of *a* and *b* Design Values

We chose to cover the ranges of our two factors with 11 points (levels) each, and generated a 121 DP full factorial design. A full factorial design tests all possible combinations, so this one contains 11×11=121 design points. Figure 39 shows the correlations and scatterplot matrix of the design. Since this is a full factorial design, we achieve perfectly even coverage over the space, and there is zero correlation between the two factors. Each DP is replicated 100 times. The output distributions of these runs are presented in Figure 40.

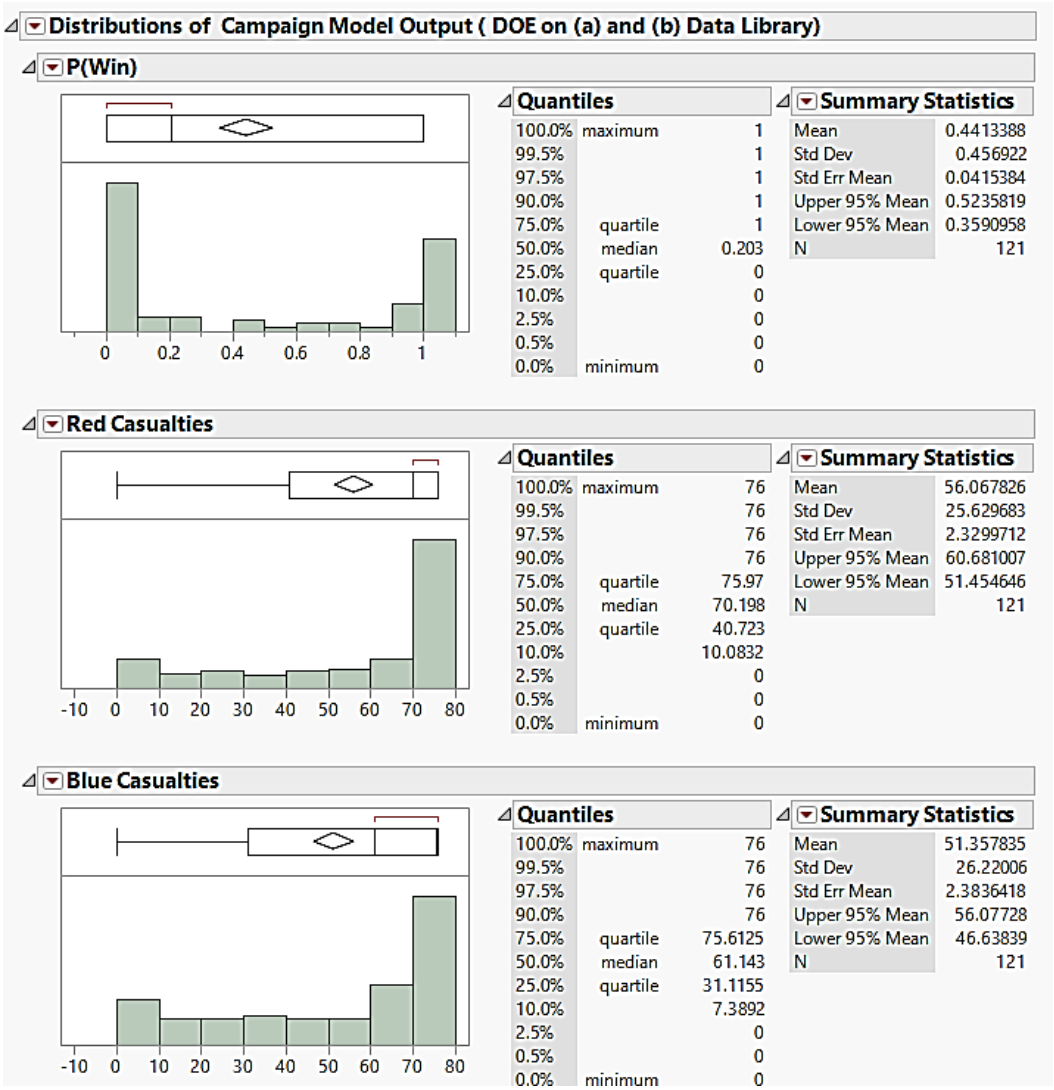


Figure 40. Distributions of Campaign Model Output (DOE on (*a*) and (*b*) Data Library)

We note that the P(Win) histogram in Figure 40 is similar to the P(Win) distribution of linking summarized NOLH data library, given in Figure 38. In the side-by-side boxplots we will present in section B.4 of this chapter, we'll continue the comparison of P(Win) and casualties to the other methods. The main goal of this method, though, is to obtain a metamodel for P(Win) as a function of *a* and *b*, which we accomplish in the next section.

(2) Fitting a Metamodel for P(Win) as a Function of *a* and *b*

With the output from the experiment, we fit a logistic regression model for P(Win) as a function of *a* and *b*.

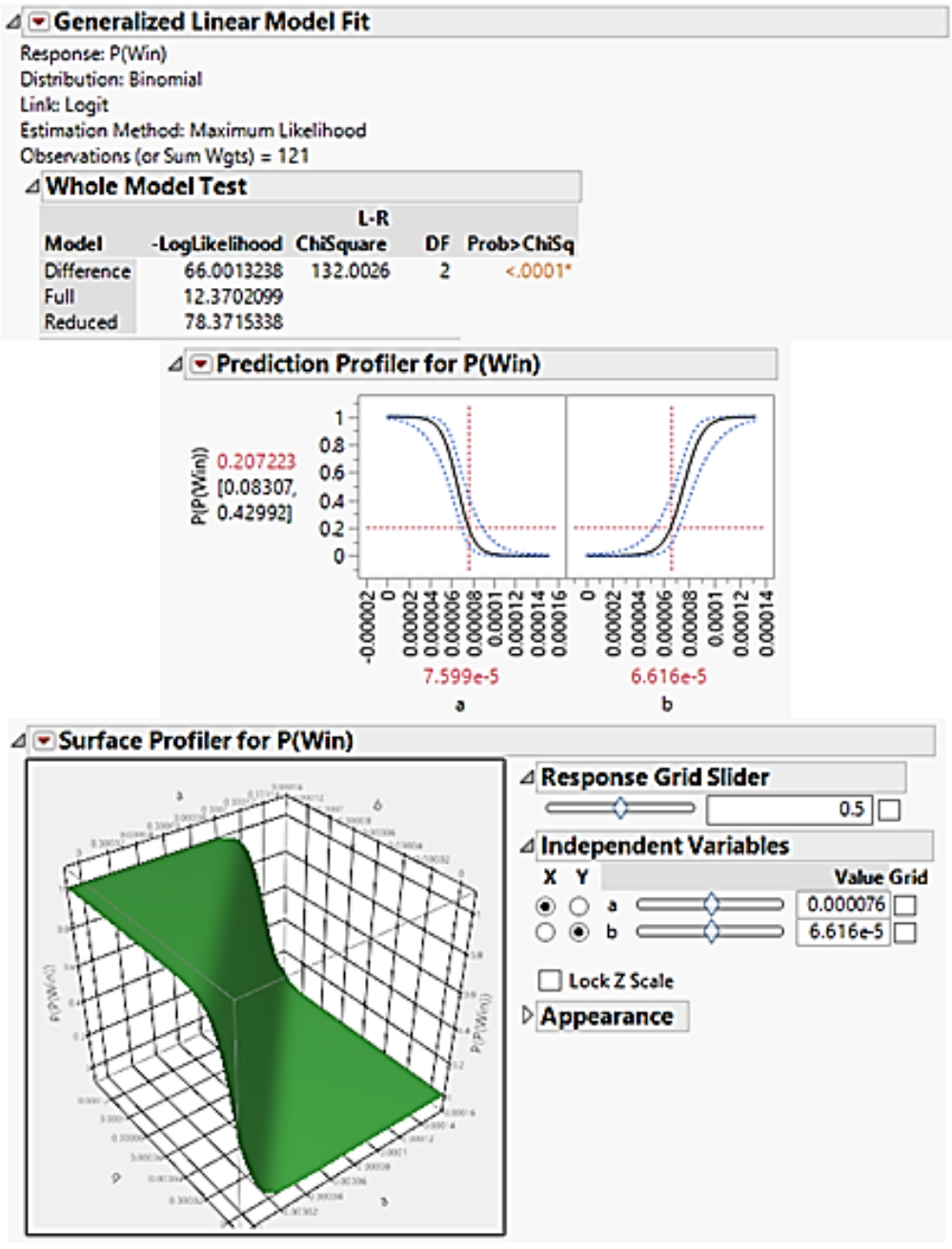


Figure 41. Logistic Regression Analysis of P(Win) = f(*a*,*b*)

We choose logistic, vice linear, regression because of the fact that our response P(Win) is constrained to the interval [0,1]. In Figure 41, the p-value for the Whole Model Test is less than our chosen level of significance (0.05), which indicates that the model is statistically significant as a whole. The Prediction Profiler shows the marginal s-shaped curves that result from logistic regression. The Surface Profiler allows us to visualize how the attrition coefficients together affect P(Win). As *a* increases, P(Win) decreases; and as *b* increases, P(Win) decreases. The prediction formula for P(Win) is shown in Figure 42.

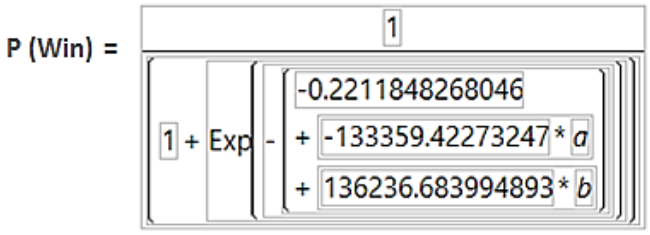


Figure 42. P(Win) Prediction Formula

(3) Construct a Composite (Embedded) Metamodel for P(Win) (Block 2D.3 of Work Flow Diagram)

We now analytically construct a formula for P(Win) = f(Blue and Red experiment variables) by plugging in the previously obtained stepwise regression models for *a* and *b* into our logistic regression of P(Win). As a result, we obtain a composite, embedded metamodel; and its formula is shown in Figure 43.

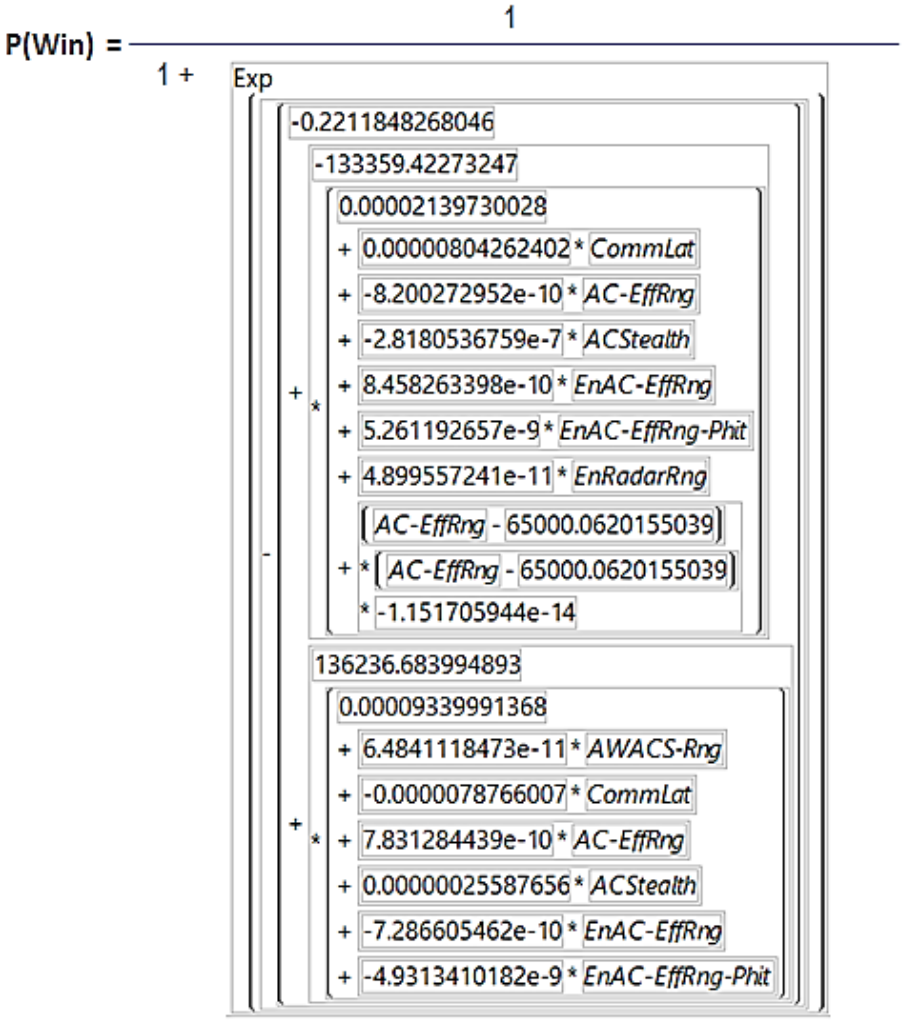


Figure 43. Embedded P(Win) Metamodel as a Function of Red and Blue Parameters

As a result, we obtain the ability to calculate an estimate for P(Win) for any combination of the engineering factors in the original MANA experiments, though it is important to note that this constructed embedded metamodel is now deterministic. Plug in a set of values for the engineering factors, and a single point estimate for P(Win) is obtained. We will next compare the performance of this metamodel to the linked version of the metamodel.

3. Comparing the Directly-Linked Metamodel to the Embedded Metamodel (Block 3 of Work Flow Diagram)

Here the main goal is to compare the result of the Linked Metamodel to the result of the Embedded Metamodel. Additionally, we compare these two to the P(Win) estimate obtained from the MANA engagement level model. Using the NOLH summary data, we will compare these three methods for estimating P(Win).

The "P(Win) Campaign Model (Embedded Meta-Model)" results are obtained by substituting each of the 129 Red and Blue experiment factor settings into the composite embedded metamodel shown in Figure 43. The correlation and scatterplot matrices appear in Figure 44. The "P(Win) Campaign Model (Linked Meta-Model)" results are obtained as described in section B.2.c.

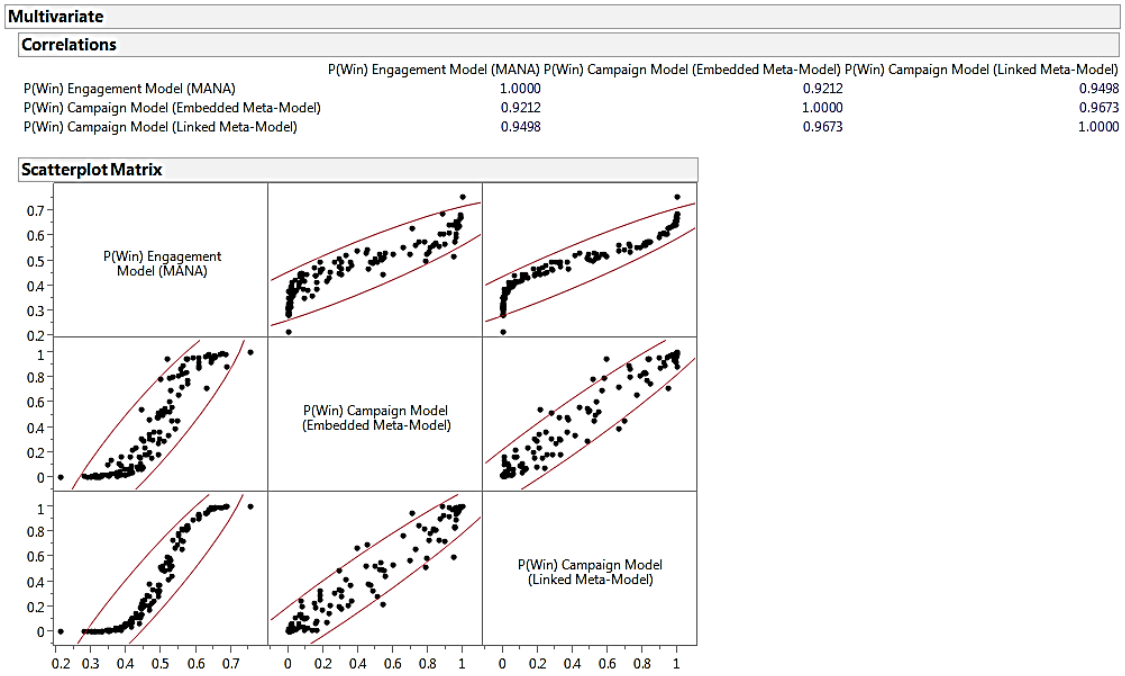


Figure 44. Comparison of Model Outputs (Correlation Matrix and Scatter Plot Matrix)

We see from Figure 44 that all pairwise correlations amongst the three are close to 1.0, which means they are nearly linearly related. Additionally, they are all positive, which is a good sign. However, we also observe how bimodal the two campaign model

results are. Since this comparison is made on the summary (means) NOLH data, we next conduct a comparison based on the full dataset.

For ease of visualization, we conduct the next comparison of the Embedded Metamodel result to the Linked result by means of a small one-factor experiment. In this experiment we will vary one of the most influential factors from our original 10-factor experiment, the effective range of the Red aircraft missile (AC-EffRng). The other factors are not varied and holding them at their baseline levels. We vary AC-EffRng from 50,000m to 80,000m, in 500m increments. Each of our resulting 61 design points is replicated 1,000 times. We first perform the MANA runs for this small experiment, to generate the 61,000 row MANA library of output. As before, to the MANA results we add columns for our calculations of the *a* and *b* estimates. We then run the stochastic Lanchester model for each of these 61,000 pairs of *a* and *b* values, replicating 100 times for each pair. As a result of the 61,000 × 100 Lanchester campaign model runs, we obtain "P(Win) Campaign Model (Linked MetaModel)."

Next, we obtain "P(Win) Campaign Model (Embedded MetaModel)" by simply evaluating the deterministic composite embedded formula for P(Win) over the range of AC-EffRng values, holding the other variables in the formula at their baseline levels.

In Figure 45, we compare the P(Win) Campaign Model (Embedded MetaModel) results in blue to the P(Win) Campaign Model (Linked MetaModel) results in red. The red dots represent the mean for each DP and the red error bars correspond to the standard deviation of the data. Because the error bar represents the standard deviation and not the confidence interval for the mean, the outermost edges of the bars fall below zero and above one. The blue dots and connecting line represent the output of the embedded metamodel. There is only one dot for each design point, because the model is deterministic. The difference in the two sets of results is evident. Clearly the use of the deterministic embedded metamodel is inadequate to assess risk.

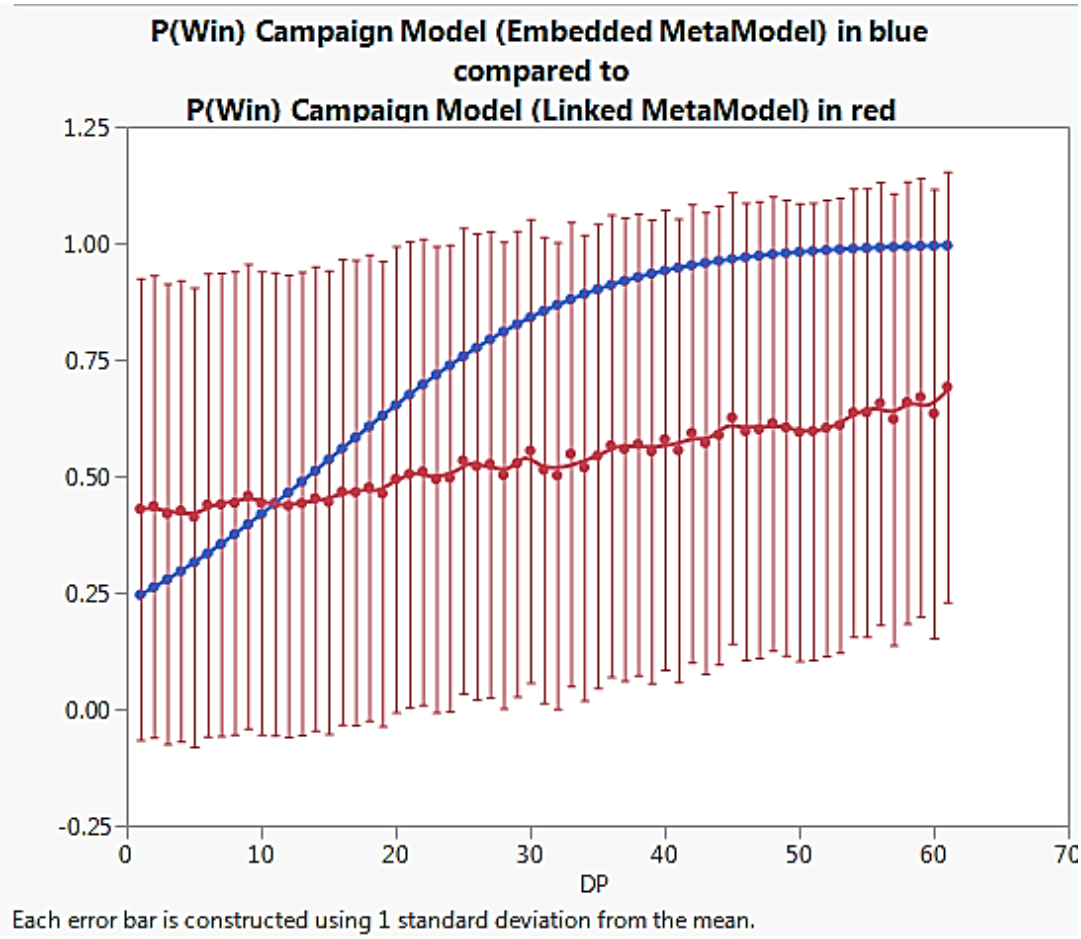


Figure 45. Comparison of Embedded to Linked Metamodel Results, Based on a One-Factor Experiment

Additionally, we note that after about design point-10 (DP-10), the P(Win) estimate obtained from embedded metamodel becomes larger than the P(Win) estimate from the linked metamodel. The use of the embedded metamodel would therefore cause an underestimation of risk for most cases.

4. Compare Campaign MOEs Based on the Sampling Methods (Block 4 of Work Flow Diagram)

We now compare the outputs of the campaign-level model sampling methods using JMP's graphic builder platform. The seven methods we have used in this research are referred to by the following names:

1. Taking Means_NOLH
2. Taking Means_R5FF

3. Random_from_NOLH
4. Random_from_R5FF
5. DOE_Campaign
6. Raw_Linked_Campaign
7. Raw_Summarized_Campaign

In Figure 46, we compare the estimate for P(Win) by method and in Figure 48, we compare the estimate for average casualties by method. Each of these figures contains side by side boxplots, with the data points overlaid. We can clearly see a large and practical impact of the sampling method chosen.

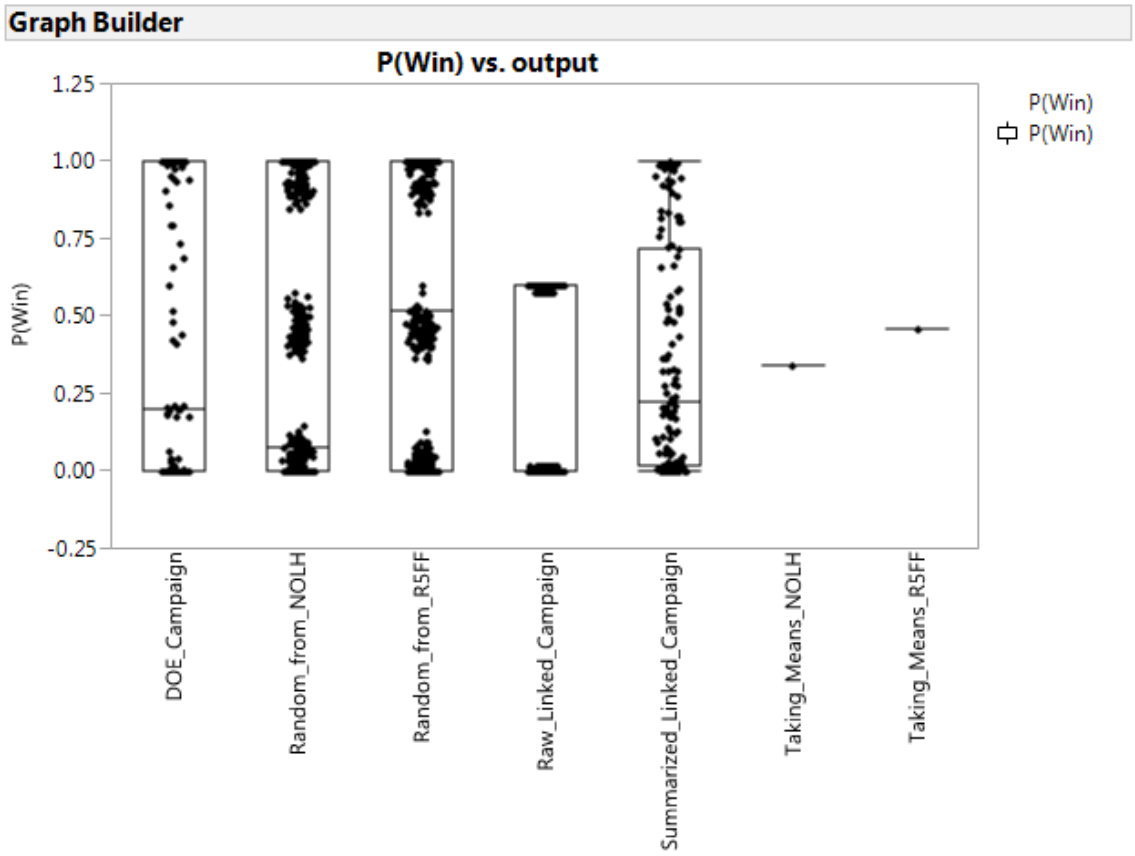


Figure 46. Comparison of Campaign P(Win) Obtained from Different Sampling Methods and Experimental Design

In Figure 46, it can be seen that the campaign-model P(Win) estimate's median and distribution differs across the sampling methods. "DOE_Campaign" and "Summarized_Linked_Campaign" have distributions that are not too dissimilar, but

Summarized_Linked_Campaign has the most even spread of points (few "gaps"). Interestingly, the "Random_from_R5FF" method resulted in output that is symmetric. The Random_from_NOLH and Random_from_R5FF approaches produced tri-modal data, and as we observed, previously, the data from Raw_Linked_Campaign is bimodal. Additionally, it is obvious that the point estimates obtained by the "Taking_Means" approaches would underestimate the variability, and thus risk, in the P(Win) campaign MOE.

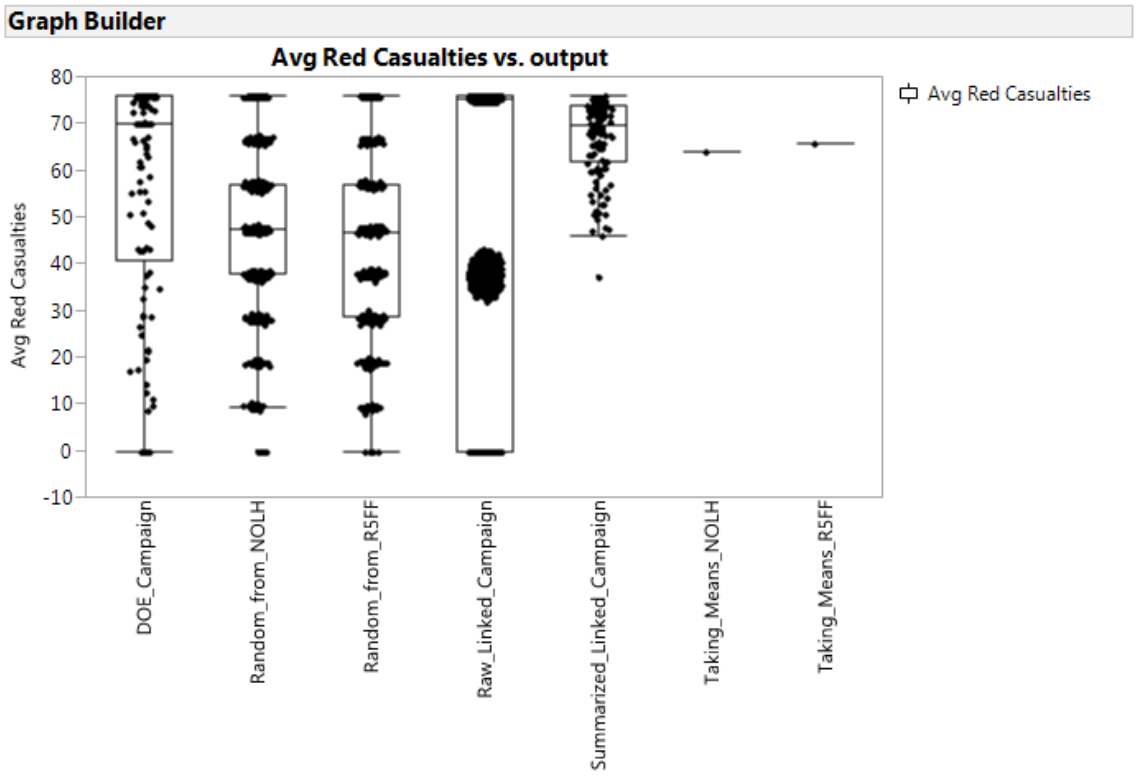


Figure 47. Comparison of Campaign Average Number of Casualties Obtained from Different Sampling Methods and Experimental Design

Figure 47 contains the boxplots overlaid with points for the average number of casualties by sampling method. In general, we expect the casualty boxplots to have skewed distributions, however, we observe close to symmetric data in the Random_from_NOLH and Random_from_R5FF cases. This may be an artifact of

sampling method. On the other hand, the Summarized_Linked_campaign data is somewhat skewed but still exhibits fairly even spread over its range, in terms of the lack of gaps. Raw_Linked_Campaign data is tri-modal, and the DOE_campaign is also somewhat skewed, but doesn't contain large gaps in its range. The Random_from_NOLH and Random_from_R5FF data exhibits gaps across their ranges, due to the nature of the sampling. Of course, the main takeaway from Figures 46 and 47 is that the campaign estimate varies substantially by sampling method.

V. DISCUSSION AND CONCLUSION

In this study, we constructed two models, one engagement-level and one campaign-level, in order to construct a simple hierarchical modeling process. Using these models, and exploring the link between the two, we explored how the variance (error) propagates through hierarchal air combat models. As mentioned in Chapter I, this thesis has sought to extend similar research conducted by LT Russell Pav in 2015. One of Pav's conclusions was that random sampling may not mitigate bias neither in the mean nor variance of campaign output (Pav, 2015). Beyond conducting random sampling on a different scenario than the one Pav used, this study employed additional techniques to sample from or link to the lower level engagement model. In particular, we explored and compared the embedded metamodeling and linked metamodeling approaches, and established a key difference between the two; namely that the deterministic embedded metamodeling technique can lead to biased results and underestimation of risk.

A. ASSESSING UNCERTAINTY AND RISK

There is always uncertainty in the output of stochastic models; that is the nature of modeling with random elements. A useful endeavor, then, is to utilize design of experiments to discover robust configurations of controllable decision variables, given uncertainty in uncontrollable variables such as the environment and enemy composition and tactics. Potentially exacerbating the uncertainty in hierarchical stochastic models would be to inappropriately sample from a lower-level model's outputs to feed as input to the higher-level model. Therefore, a goal of this study was to explore methods to accomplish this. We generated a MANA engagement model to provide input to a stochastic Lanchester campaign model, and performed designed experiments on the MANA model. We then analyzed engagement model outputs to analyze their distributions and fit metamodels for the MOEs as functions of the experiment variables. We then compared inputs for the campaign model obtained by a variety of sampling methods, including executing a designed experiment on the campaign model. Figure 31 and Figure 32 illustrated how the attrition coefficient distributions differed according to

method. This is the first level of assessment of uncertainty in hierarchical modelling—examining the distributions of inputs.

After running the campaign model with different input libraries, we conducted the second level of assessment of uncertainty in hierarchical modelling. Figures 46 and 47 contain the side by side box plots that show how the campaign model output can differ substantially according to the sampling method that is used. Utilizing one of the methods which is not sufficient to propagate variance from the engagement model output to the campaign model may lead the analyst to an incorrect assessment that may underestimate risk, potentially costing money and losing lives.

Using point estimation by taking the sample mean is the easiest and simplest way among these sampling methods, and is commonly used; however, Figures 46 and 47 show that taking means does not provide the ability to evaluate variability and risk. Random sampling techniques provide a distribution of results, but we should keep in mind that not all random sampling methods fully characterize the underlying uncertainty, and the result may change on repeated runs.

We also discuss two techniques which we call the deterministic embedded metamodel approach and the linked metamodel approach. As Figure 45 demonstrates, the linked metamodel is capable of characterizing the underlying uncertainty, while the deterministic embedded metamodel does not. Also, as we discussed, an excellent method for systematically exploring uncertainties in the scenario is design of experiments (DOE), and the result of the DOE can adequately characterize variability and risk. We chose to illustrate the use of DOE by using both the Nearly Orthogonal Latin Hypercube (NOLH) and the Resolution 5 Fractional Factorial. Figure 45 shows that the metamodel linked to DOE output is much more effective and useful for quantifying risk than the embedded metamodel, since it captures the variability that results when one of the engineering factors is varied over a range. Additionally, we recommended that if time permits, it would be desirable to supplement the desirable features of the space-filling NOLH with the corner-sampling R5FF.

B. AREAS FOR FUTURE RESEARCH

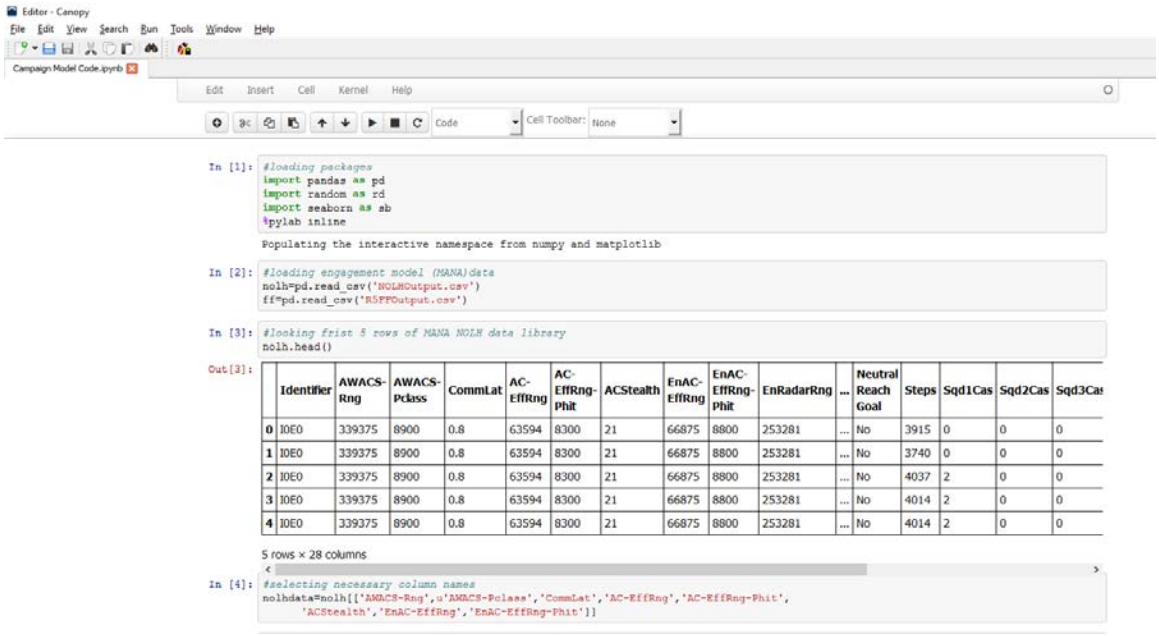
We generated a two vs. two air-to-air engagement model since no fighter aircraft flies alone. While analyzing the result of linking the raw (full) NOLH data library to the campaign model, we observed that the two vs. two engagement's outcome can be either one side wins with no casualties or one casualty. In other words, the two vs. two engagement provides less variability in model output than there would be in a several vs. several engagement, since there are only two outcomes for each side. As a result of that, campaign outputs obtained from the runs on raw (full) NOLH data library exhibited very low variability. Therefore, to achieve more variability in engagement model output, the number of fighters could be increased (three vs. three or four vs. four). This would increase the variability in the engagement level model output; and linking the raw data library to the campaign model can then be more plausible and serve as an effective way to propagate variance forward into the campaign model.

This study employed MANA for the engagement model and the Lanchester linear law for the campaign model. Even though MANA allows the user to model many engineering factors of military assets and battle conditions, the program comes with some significant limitations and assumptions. Additionally, we do not use models that are accredited by any official organization. There certainly exist accredited higher resolution simulation programs for modeling air missions. Thus, using more detailed and realistic modeling tools such as BRAWLER and STORM for studies related hierarchical combat modeling may provide better insights.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

Here, we provide the code for the stochastic Lanchester linear campaign models within a Python Notebook page. First, necessary programming packages are installed, then the model takes in csv files of both NOLH and R5FF data libraries that are obtained from MANA model runs. We built a function called “lanch” which simulates one time period of an air campaign. Then, we employed different sampling methods to sample from both data libraries to provide input to the campaign model. A campaign model code for each sampling technique was written and “lanch” function was utilized within these campaign codes to simulate 100 vs. 100 campaigns between Red and Blue. The number of replications for campaign models are arbitrarily selected to have enough runs to provide mean statistics for each campaign MOES. After each model run, summary statistics are recorded as a data frame and written to a csv file. Since Python Notebook is employed, screen shots or code and statistics are presented below in Figures.


Editor - Canopy

File Edit View Search Run Tools Window Help

Campaign Model Code.ipynb

Edit Insert Cell Kernel Help

Code Cell Toolbar: None

```
In [1]: #loading packages
        import pandas as pd
        import random as rd
        import seaborn as sb
        %pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

```
In [2]: #loading engagement model (MANA)data
        nolh=pd.read_csv('NOLHOutput.csv')
        ff=pd.read_csv('R5FFOutput.csv')
```

```
In [3]: #looking frist 5 rows of MANA NOLH data library
        nolh.head()
```

Out[3]:

	Identifier	AWACS-Rng	AWACS-Pclass	CommLat	AC-EffRng	AC-EffRng-Phit	ACStealth	EnAC-EffRng	EnAC-EffRng-Phit	EnRadarRng	...	Neutral Reach Goal	Steps	Sqd1Cas	Sqd2Cas	Sqd3Cas
0	I0E0	339375	8900	0.8	63594	8300	21	66875	8800	253281	...	No	3915	0	0	0
1	I0E0	339375	8900	0.8	63594	8300	21	66875	8800	253281	...	No	3740	0	0	0
2	I0E0	339375	8900	0.8	63594	8300	21	66875	8800	253281	...	No	4037	2	0	0
3	I0E0	339375	8900	0.8	63594	8300	21	66875	8800	253281	...	No	4014	2	0	0
4	I0E0	339375	8900	0.8	63594	8300	21	66875	8800	253281	...	No	4014	2	0	0

5 rows × 28 columns

```
In [4]: #selecting necessary column names
        nolhdata=nolh[['AWACS-Rng',u'AWACS-Pclass','CommLat','AC-EffRng','AC-EffRng-Phit',
              'ACStealth','EnAC-EffRng','EnAC-EffRng-Phit']]
```

Figure 48. Screen Shot of Campaign Model Code within Python Notebook-1

In [5]:
```
#looking first 5 rows of selected NOLH Data Library
nolhdata.head()
```

Out[5]:

	AWACS-Rng	AWACS-Pclass	CommLat	AC-EffRng	AC-EffRng-Phit	ACStealth	EnAC-EffRng	EnAC-EffRng-Phit
0	339375	8900	0.8	63594	8300	21	66875	8800
1	339375	8900	0.8	63594	8300	21	66875	8800
2	339375	8900	0.8	63594	8300	21	66875	8800
3	339375	8900	0.8	63594	8300	21	66875	8800
4	339375	8900	0.8	63594	8300	21	66875	8800

In [6]:
```
#looking first 5 rows of selected NOLH Data Library
ff.head()
```

Out[6]:

	Identifier	AWACS-Rng	CommLat	AC-EffRng	ACStealth	EnAC-EffRng	EnRadarRng	AWACS-Pclass	AC-EffRng-Phit	EnAC-EffRng-Phit	...	Neutral Reach Goal	Steps	Sqd1Cas	Sqd2Cas	Sqd3Cas
0	I0E0	400000	2	80000	30	80000	280000	10000	9000	9000	...	No	3716	0	0	0
1	I0E0	400000	2	80000	30	80000	280000	10000	9000	9000	...	No	3683	1	0	0
2	I0E0	400000	2	80000	30	80000	280000	10000	9000	9000	...	No	3902	2	0	0
3	I0E0	400000	2	80000	30	80000	280000	10000	9000	9000	...	No	3874	2	0	0
4	I0E0	400000	2	80000	30	80000	280000	10000	9000	9000	...	No	3677	0	0	0

5 rows × 28 columns

Figure 49. Screen Shot of Campaign Model Code within Python Notebook-2

In [7]:
```
#BUILDING CAMPAIGN MODEL FUNCTION
```

In [8]:
```
#Campaign includes 4 operations and that makes 4 time periods. (4 times 25 vs 25)
#Each time periods has breaek points (1/4 of opearation package size-> 6 aircraft)
#Below, there is a Stochastic Lanchester Model Campaign Model function for a time period.
#At the end of each run, it returns the number of casualties of both side.
```

In [9]:
```
def lanch(a,b):
    m=25; n=25; t=0
    while(m>6 and n>6): #breakpoints
        rate=a*n*m+b*m*n
        ttnc=rd.expovariate(rate)
        t=t+ttnc
        if(rd.random() <= a*n*m/rate):
            m=m-1
        else:
            n=n-1
    return (25-m,25-n)
```

Figure 50. Screen Shot of Campaign Model Code within Python Notebook-3

In [10]:
```
#USING DIFFIRENT SAMPLING METHODS TO GET DATA FROM NOLH AND R5FF DATA LIBRARIES TO FEED CAMPAIGN MODEL.
```

In [11]:
```
###1 TAKING SAMPLE AVERAGE(MEAN)
```

In [12]:
```
    #1a MEANS FROM NOLH DATA LIBRARY
```

In [13]:
```
#calculating Red's killing power on Blue (b)
b1 = nolh['Alleg1Cas(Blue)'].mean() /  (nolh['Steps'].mean()*2*2)
b1
```

Out[13]: 7.593350522117206e-05

In [14]:
```
#calculating Blue's killing power on Red (a)
a1 = nolh['Alleg2Cas(Red)'].mean() /  (nolh['Steps'].mean()*2*2)
a1
```

Out[14]: 8.096973253228005e-05

In [15]:
```
#running Campaign Model
#10,000 replications using overall mean of casualties and timestep
```

In [16]:
```
x_win=0
xcasulty=[]
ycasulty=[]
replications=10000
for i in range(replications):
    x=0
    y=0 #x red, y blue
    for j in range(4):
        temp = lanch(a1,b1)
        x=x+temp[0]
        y=y+temp[1]
    x_win = x_win+(x<y)
    xcasulty.append(x)
    ycasulty.append(y)
    p_x_win= x_win *0.0001
```

Figure 51. Screen Shot of Campaign Model Code within Python Notebook-4

In [17]:
```
#Probabiltiy Red Wins
p_x_win
```

Out[17]: 0.34290000000000004

In [18]:
```
#building a dataframe with pandas package
output1_Taking_Means_Nolh = pd.DataFrame({'Red Casualties': xcasulty, 'Blue Casualties': ycasulty })
```

In [19]:
```
#looking at first 5 rows of Campaign output
output1_Taking_Means_Nolh.head()
```

Out[19]:

	Blue Casualties	Red Casualties
0	68	65
1	58	68
2	66	70
3	68	66
4	65	70

In [20]:
```
#writing output data to a csv file
output1_Taking_Means_Nolh.to_csv('output1_Taking_Means_Nolh.csv')
```

In [21]:
```
    #1b MEANS FROM R5FF DATA LIBRARY
```

In [22]:
```
#calculating Red's killing power on Blue (b)
b2 = ff['Alleg1Cas(Blue)'].mean() /  (ff['Steps'].mean()*2*2)
b2
```

Out[22]: 7.997342414462116e-05

In [23]:
```
#calculating Blue's killing power on Red (a)
a2= nolh['Alleg2Cas(Red)'].mean() /  (nolh['Steps'].mean()*2*2)
a2
```

Figure 52. Screen Shot of Campaign Model Code within Python Notebook-5

In [24]:
```
#Running campaign model
#10,000 replications using overall mean of casualties and timestep
```

In [25]:
```
x_win=0
xcasulty=[]
ycasulty=[]
replications=10000
for i in range(replications):
    x=0
    y=0 #x red, y blue
    for j in range(4):
        temp = lanch(a2,b2)
        x=x+temp[0]
        y=y+temp[1]
    x_win = x_win+(x<y)
    xcasulty.append(x)
    ycasulty.append(y)
    p_x_win= x_win *0.0001
```

In [26]:
```
#probabiltiy of Red Wins
p_x_win
```

Out[26]:
```
0.45430000000000004
```

In [27]:
```
#building a dataframe with pandas package
output2_Taking_Means_R5FF = pd.DataFrame({'Red Casualties': xcasulty, 'Blue Casualties': ycasulty })
```

Figure 53. Screen Shot of Campaign Model Code within Python Notebook-6

In [28]:
```
#looking at first 5 rows of Campaign output
output2_Taking_Means_R5FF.head()
```

Out[28]:

	Blue Casualties	Red Casualties
0	60	68
1	67	72
2	73	66
3	54	76
4	67	68

In [29]:
```
#writing output data to a csv file
output2_Taking_Means_R5FF.to_csv('output2_Taking_Means_R5FF.csv')
```

Figure 54. Screen Shot of Campaign Model Code within Python Notebook-7

In [30]:
```
###2 RANDOM SMAPLING
```

In [31]:
```
#1000 runs, where each run consists of using 4 different a values and 4 different b values. 4 x 1000 = 4000 a and b
```

In [32]:
```
    #2a RANDOM SAMPLING FROM NOLH DATA LIBRARY
```

In [33]:
```
nolh_1000=list() #creating list to save data
for i in range(1000):
    sp=rd.sample(range(129000),4) #generating 4 random number between 0 and 129.000
    #calculating a and b
    a_list=list(nolh.ix[sp]['Alleg2Cas(Red)'] / (nolh.ix[sp]['Steps'] *2*2))
    b_list=list(nolh.ix[sp]['Alleg1Cas(Blue)'] / (nolh.ix[sp]['Steps'] *2*2))

    x_win=0 #initiating number of x wins
    #generating lists for casualties
    xcasulty=[]
    ycasulty=[]
    for i in range(100):
        x=0
        y=0 #x red, y blue
        for j in range(4):
            temp = lanch(a_list[j],b_list[j]) #running campaign model function
            x=x+temp[0]
            y=y+temp[1]
        x_win = x_win+(x<y) #determining the winner of each campaign (4 period)
        xcasulty.append(x)
        ycasulty.append(y)
    #calculating winning probabilities
    prob_x_win = x_win/ 100.0
    prob_y_win = 1- prob_x_win
    #Saving data
    full_list=a_list+b_list+[x_win,prob_x_win, prob_y_win,mean(xcasulty),mean(ycasulty)]
    nolh_1000.append(full_list)
```

Figure 55. Screen Shot of Campaign Model Code within Python Notebook-8

In [34]:
```
#converting list to a dataframe , so I can plot and see data more easily.
random_from_NOLH= pd.DataFrame(nolh_1000,columns=['a1','a2','a3','a4','b1','b2','b3','b4','nxw','prob_x_win', 'prob_y_win','mean_x_
```

In [35]:
```
random_from_NOLH.head()
```

Out[35]:

	a1	a2	a3	a4	b1	b2	b3	b4	nxw	prob_x_win	prob_y_win	mean_x_cas	mean_y_cas
0	0.000129	0.000130	0.000136	0.000138	0.000064	0.000065	0.000000	0.000000	0	0.0	1.0	75.88	19.48
1	0.000132	0.000000	0.000133	0.000131	0.000000	0.000127	0.000000	0.000065	0	0.0	1.0	56.92	28.67
2	0.000000	0.000000	0.000051	0.000131	0.000125	0.000125	0.000102	0.000066	100	1.0	0.0	29.14	65.99
3	0.000000	0.000000	0.000136	0.000063	0.000126	0.000122	0.000000	0.000127	100	1.0	0.0	28.91	56.93
4	0.000129	0.000134	0.000135	0.000062	0.000065	0.000000	0.000000	0.000124	0	0.0	1.0	66.40	28.57

In [36]:
```
#writing output data to a csv file
random_from_NOLH.to_csv('output3_random_from_NOLH.csv')
```

Figure 56. Screen Shot of Campaign Model Code within Python Notebook-9

In [37]:
```
    #2B RANDOM SAMPLING FROM R5FF DATA LIBRARY
```

In [38]:
```
ff_1000=list()
for i in range(1000):
    sp=rd.sample(range(128000),4)
    a_list=list(ff.ix[sp]['Alleg2Cas(Red)'] / (ff.ix[sp]['Steps'] *2*2))
    b_list=list(ff.ix[sp]['Alleg1Cas(Blue)'] / (ff.ix[sp]['Steps'] *2*2))

    x_win=0
    xcasulty=[]
    ycasulty=[]
    for i in range(100):
        x=0
        y=0 #x red, y blue
        for j in range(4):
            temp = lanch(a_list[j],b_list[j])
            x=x+temp[0]
            y=y+temp[1]
        x_win = x_win+(x<y)
        xcasulty.append(x)
        ycasulty.append(y)
    prob_x_win = x_win/ 100.0
    prob_y_win = 1- prob_x_win
    full_list=a_list+b_list+[x_win,prob_x_win, prob_y_win,mean(xcasulty),mean(ycasulty)]
    ff_1000.append(full_list)
```

In [39]:
```
random_from_R5FF= pd.DataFrame(ff_1000,columns=['a1','a2','a3','a4','b1','b2','b3','b4','nxw','prob_x_win', 'prob_y_win','mean_x_ca
```

Figure 57. Screen Shot of Campaign Model Code within Python Notebook-10

In [40]:
```
random_from_R5FF.head()
```

Out[40]:

	a1	a2	a3	a4	b1	b2	b3	b4	nxw	prob_x_win	prob_y_win	mean_x_cas	mean_y_cas
0	0.000000	0.000141	0.000000	0.000064	0.000103	0.000000	0.000128	0.000127	100	1.0	0.0	28.13	56.98
1	0.000000	0.000061	0.000128	0.000000	0.000123	0.000123	0.000000	0.000120	100	1.0	0.0	28.52	56.99
2	0.000000	0.000063	0.000000	0.000140	0.000129	0.000126	0.000123	0.000000	100	1.0	0.0	28.40	56.99
3	0.000052	0.000000	0.000065	0.000138	0.000105	0.000127	0.000130	0.000000	100	1.0	0.0	37.62	56.97
4	0.000136	0.000135	0.000051	0.000137	0.000068	0.000000	0.000103	0.000068	0	0.0	1.0	66.42	37.39

In [41]:
```
random_from_R5FF.to_csv('output4_random_from_R5FF.csv')
```

In [42]:
```
#3 EMBEDDED METAMODELING
```

In [43]:
```
#Perform a DOE on the lanchester model, varying a and b over ranges informed by the R5FF data library summary statitics.
```

In [44]:
```
ab_design=pd.read_csv('a_b_designpoint.csv')
```

In [45]:
```
ab_design.head(5) #Looking at DPs of a and b
```

Out[45]:

	a	b
0	1.000000e-10	1.000000e-10
1	1.519770e-05	1.000000e-10
2	3.039520e-05	1.000000e-10
3	4.559280e-05	1.000000e-10
4	6.079030e-05	1.000000e-10

Figure 58. Screen Shot of Campaign Model Code within Python Notebook-11

In [46]:
```
ab_DP_Campaign=list()
for i in range(len(ab_design)):
    a = ab_design.ix[i]['a']
    b = ab_design.ix[i]['b']
    x_win=0
    prob_x_win= 0
    xcasulty=[]
    ycasulty=[]
    for j in range(1000):
        x=0
        y=0 #x red, y blue
        for k in range(4):
            temp = lanch(a,b)
            x=x+temp[0]
            y=y+temp[1]
        x_win = x_win+(x<y)
        xcasulty.append(x)
        ycasulty.append(y)
    prob_x_win = x_win/ 1000.0
    prob_y_win = 1- prob_x_win
    full_list=[a,b,x_win,prob_x_win, prob_y_win, mean(xcasulty),mean(ycasulty)]
    ab_DP_Campaign.append(full_list)
```

In [47]:
```
ab_DP_Campaign=pd.DataFrame(ab_DP_Campaign,columns=['a','b','x_win','prob_x_win','prob_y_win','mean(xcasulty)','mean(ycasulty)'])
```

In [48]:
```
ab_DP_Campaign.columns = ['a','b','x_win','prob_x_win','prob_y_win','mean_xcasulty','mean_ycasulty']
```

Figure 59. Screen Shot of Campaign Model Code within Python Notebook-12

In [49]:
```
ab_DP_Campaign.head(5)
```

Out[49]:

	a	b	x_win	prob_x_win	prob_y_win	mean_xcasulty	mean_ycasulty
0	1.000000e-10	1.000000e-10	477	0.477	0.523	66.576	66.226
1	1.519770e-05	1.000000e-10	0	0.000	1.000	76.000	0.000
2	3.039520e-05	1.000000e-10	0	0.000	1.000	76.000	0.000
3	4.559280e-05	1.000000e-10	0	0.000	1.000	76.000	0.001
4	6.079030e-05	1.000000e-10	0	0.000	1.000	76.000	0.000

In [50]:
```
ab_DP_Campaign.to_csv('output5_ab_DP_Campaign.csv')
```

In [51]:
```
###4LINKING ENGAGEMENT MODEL AND CAMPAIGN MODEL DIRECTLY (Linked Metamodeling)
```

In [52]:
```
    #4a Running 10 replicationS for each experiment outputs in raw NOLH data library (129,000)
#Takes more time to run since 129.000 x 10 = 1,290,000
```

In [53]:
```
ab_NOLH_Raw=pd.read_csv('NOLHOutput_appended.csv')
```

In [54]:
```
ab_NOLH_Raw.head()
```

Out[54]:

	Identifier	DP	AWACS-Rng	AWACS-Pclass	CommLat	AC-EffRng	AC-EffRng-Phit	ACStealth	EnAC-EffRng	EnAC-EffRng-Phit	...	Sqd2Cas	Sqd3Cas	Sqd4Cas	Sqd1Inj	Sqd2Inj	Sqd3
0	I0E0	1	339375	8900	0.8	63594	8300	21	66875	8800	...	0	0	2	0	0	0
1	I0E0	1	339375	8900	0.8	63594	8300	21	66875	8800	...	0	0	2	0	0	0
2	I0E0	1	339375	8900	0.8	63594	8300	21	66875	8800	...	0	0	1	0	0	0
3	I0E0	1	339375	8900	0.8	63594	8300	21	66875	8800	...	0	0	0	0	0	0

Figure 60. Screen Shot of Campaign Model Code within Python Notebook-13

In [55]:
```
ab_NOLH_Raw_Campaign=list()
replications= 30
for i in range(len(ab_NOLH_Raw)):
    a = ab_NOLH_Raw.ix[i]['a']
    b = ab_NOLH_Raw.ix[i]['b']
    x_win=0
    prob_x_win= 0
    xcasulty=[]
    ycasulty=[]
    for j in range(replications):
        x=0
        y=0 #x red, y blue
        for k in range(4):
            temp = lanch(a,b)
            x=x+temp[0]
            y=y+temp[1]
        x_win = x_win+(x<y)
        xcasulty.append(x)
        ycasulty.append(y)
    prob_x_win = x_win/ 50.0
    prob_y_win = 1- prob_x_win
    full_list=[a,b,x_win,prob_x_win,  prob_y_win, mean(xcasulty),mean(ycasulty)]
    ab_NOLH_Raw_Campaign.append(full_list)
```

In [56]:
```
ab_NOLH_Raw_Campaign=pd.DataFrame(ab_NOLH_Raw_Campaign,columns=['a','b','x_win','prob_x_win','prob_y_win','mean(xcasulty)','mean(yc
```

In [57]:
```
ab_NOLH_Raw_Campaign.columns = ['a','b','x_win','prob_x_win','prob_y_win','mean_xcasulty','mean_ycasulty']
```

In [58]:
```
ab_NOLH_Raw_Campaign.head(5)
```

Out[58]:

	a	b	x_win	prob_x_win	prob_y_win	mean_xcasulty	mean_ycasulty
0	0.000128	0.000000	0	0.0	1.0	76.000000	0.000000
1	0.000134	0.000000	0	0.0	1.0	76.000000	0.000000
2	0.000062	0.000124	30	0.6	0.4	39.833333	75.666667

Figure 61. Screen Shot of Campaign Model Code within Python Notebook-14

```
In [59]: ab_NOLH_Raw_Campaign.to_csv('output6_Raw_Linked_Campaign.csv')

In [60]:     #4B USING SUMMARY STATISTICS TO CALCULATE A AND B AND RUNNING THEM

In [61]: #For each of the 129 design points, calculate the a and b for that design point,
         #based on the average red and blue casualties over the 1000 reps. Run the stochastic lanchester model
         #129 x 1000 times, where for each run, you use the same a and b for all time periods and all 1000 runs.

In [62]: ab_NOLH_Summary=pd.read_csv('NOLHOutput_appended_summary.csv')

In [63]: ab_NOLH_Summary.head(5)
```

Out[63]:

	Identifier	DP	AWACS-Rng	AWACS-Pclass	CommLat	AC-EffRng	AC-EffRng-Phit	ACStealth	EnAC-EffRng	EnAC-EffRng-Phit	...	Mean(Steps)	Mean a	Mean b	P(Win)	Pred Formula Mean(a (killing power of blue on red))	P F M (I p o o b
0	I0E0	1	339375	8900	0.8	63594	8300	21	66875	8800	...	3942.573	0.000083	0.000074	0.472	0.000085	0
1	I0E1	2	391250	8600	0.9	63828	8100	14	62656	8200	...	3849.768	0.000077	0.000084	0.545	0.000081	0

Figure 62. Screen Shot of Campaign Model Code within Python Notebook-15

```
In [64]: ab_NOLH_Summary_Campaign=list()
         for i in range(len(ab_NOLH_Summary)):
             a = ab_NOLH_Summary.ix[i]['Mean a']
             b = ab_NOLH_Summary.ix[i]['Mean b']
             x_win=0
             prob_x_win= 0
             xcasulty=[]
             ycasulty=[]
             for j in range(1000):
                 x=0
                 y=0 #x red, y blue
                 for k in range(4):
                     temp = lanch(a,b)
                     x=x+temp[0]
                     y=y+temp[1]
                 x_win = x_win+(x<y)
                 xcasulty.append(x)
                 ycasulty.append(y)
             prob_x_win = x_win/ 1000.0
             prob_y_win = 1- prob_x_win
             full_list=[a,b,x_win,prob_x_win,  prob_y_win, mean(xcasulty),mean(ycasulty)]
             ab_NOLH_Summary_Campaign.append(full_list)
```

Figure 63. Screen Shot of Campaign Model Code within Python Notebook-16

```
In [65]: ab_NOLH_Summary_Campaign=pd.DataFrame(ab_NOLH_Summary_Campaign,columns=['a','b','x_win','prob_x_win','prob_y_win','mean(xcasulty)',
         <                                                                                                                              >

In [66]: ab_NOLH_Summary_Campaign.columns = ['a','b','x_win','prob_x_win','prob_y_win','mean_xcasulty','mean_ycasulty']

In [67]: ab_NOLH_Summary_Campaign.head()
```

Out[67]:

	a	b	x_win	prob_x_win	prob_y_win	mean_xcasulty	mean_ycasulty
0	0.000083	0.000074	244	0.244	0.756	69.913	61.634
1	0.000077	0.000084	672	0.672	0.328	63.306	68.902
2	0.000101	0.000054	0	0.000	1.000	75.567	40.477
3	0.000102	0.000055	0	0.000	1.000	75.651	40.629
4	0.000071	0.000090	907	0.907	0.093	57.215	72.315

```
In [68]: ab_NOLH_Summary_Campaign.to_csv('output7_Summarized_Linked_Campaign.csv')

In [69]: #End of Campaign Model code.
```

Figure 64. Screen Shot of Campaign Model Code within Python Notebook-17

LIST OF REFERENCES

Asiribo, O., & Gurland, J. (1990). Coping with variance heterogeneity. *Communications in Statistics–Theory and Methods, 19*(11), 4029–4048.

Barton, R. R. (1998). *Simulation metamodels*. In *Proceedings of the 30th conference on Winter Simulation* (pp. 167–176). IEEE Computer Society Press.

Box, G. E. (1987). *Empirical model-building and response surfaces.* New York: John Wiley & Sons.

Brown, D. (1987). *The Royal Navy and Falklands War*. London: Pen and Sword.

Brown, M. B., & Forsythe, A. B. (1974). Robust tests for the equality of variances. *Journal of the American Statistical Association, 69*(346), 364–367.

Cerniglia-Mosher, M. (2009) Retrieved from Air Force Agency for Simulation Modeling: Retrieved from: http://www.afams.af.mil/shared/media/document/AFD-090416-077.pdf

Chant, C. (2001). *Air war in the Falklands 1982* (Vol. 28). Oxford: Osprey Publishing.

Cioppa, T. M., & Lucas, T. W. *(2007).* Efficient nearly orthogonal and space-filling Latin hypercubes. *Technom*etrics, *49*(1), 45–55.

Defense Systems Information Analysis Center. (2015). Brawler. Retrieved from https://www.dsiac.org/resources/models_and_tools/brawler

Engel, J. H. (1954). A verification of Lanchester's law. *Journal of the Operations Research Society of America, 2*(2), 163–171.

Hartley III, D. S. (2001). Battle modelingBATTLE MODELING. *In Encyclopedia of Operations Research and Management Science* (pp. 53–57). New York: Springer.

Horwood, I., Mackay, N., & Price, C. (2014). *Concentration and asymmetry in air combat: Lessons for the defensive employment of air power.* York, UK.: University of York.

Jane's All the World's Aircraft. (2016). Lockheed Martin F-35 Lightning II. Retrieved from https://janes.ihs.com/Janes/Display/1343368

Kleijnen, J. P., Sanchez, S. M., Lucas, T. W., & Cioppa, T. M. (2005). State-of-the-art review: A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing*, 17(3), 263–289.

Lanchester, F. W. (1916). *Aircraft in Warfare: The Dawn of the Forth Arm.* Edited by J.R. Newman and Simon and Schester, Vol. 4. Charleston, SC: BiblioLife.

Lawson, A. B. (2013). *Bayesian disease mapping: Hierarchical modeling in spatial epidemiology*. CRC press.

Lucas, T. W. (2000). The Stochastic Versus Deterministic Argument for Combat Simulations: Tales of when the average won't do. *Military Operations Research*, *5*(3), 9–28.

Lucas, T. W., & Dinges, J. A. (2004). The effect of battle circumstances on fitting Lanchester equations to the Battle of Kursk. *Military Operations Research*, *9*(2), 17–30.

Lucas, T. W., & McGunnigle, J. E. (2003). When is model complexity too much? Illustrating the benefits of simple models with Hughes' salvo equations. *Naval Research Logistics (NRL), 50*(3), 197–217.

Lucas, T. W., & Turkes, T. (2004). Fitting Lanchester equations to the battles of Kursk and Ardennes. *Naval Research Logistics (NRL), 51*(1), 95–116.

McIntosh, G. C. (2009). *MANA-V (Map aware non-uniform automata – Vector) supplementary manual.* Auckland, New Zealand: New Zealanf Defence Technology Agency.

Pape, R. A. (2004). The true worth of air power. *Foreign Affairs, 83*(2), 116–130. Retrieved from https://www.foreignaffairs.org/articles/2004-03-01/true-worth-air-power

Pav, R. (2015). *Experiments in Error Propagation Within Hierarchical Combat Models.* Master's thesis, Naval Postgraduate School, Monterey, CA.

Rao, A., Lucas, A., Morley, D., Selvestrel, M., & Murray, G. (1993). *Agent-Oriented Architecture for Air-Combat Simulation.* Melbourne, Australia: The Australian Artificial Intellegence Institute.

Sanchez, S. M., Lucas, T. W., Sanchez, P. J., Nannini, C. J., & Wan, H. (2012). "Designs for large-scale simulation experiments,with applications to defense and homeland security." In *Design and Analysis of Experiments*, Volume 3, edited by K. Hinkelmann, 413–441. Wiley: New York.

Savage, S. (2002, November). The flaw of averages. *Harvard Business Review*, 4.

Seymour, C. N. (2014). Capturing the full potential of the Synthetic Theater Operations Research Model (STORM). Master's thesis, Naval Postgraduate School, Monterey, CA. Retrieved from Calhoun: http://calhoun.nps.edu /handle/10945/44000

U.S. Air Force. (2015, Sep 23). Retrieved from F-16 Fighting Falcon: http://www.af.mil/AboutUs/FactSheets/Display/tabid/224/Article/104505/f-16-fighting-falcon.aspx

United States Government Accountability Office. (2004, May 3). *Joint Strike Fighter Acquisition.* Retrieved from U.S. Government Accountability Office: http://www.gao.gov/new.items/d04554.pdf

Wackerly, D., Mendenhall, W., & Scheaffer, R. (2007). *Mathematical statistics with applications,* 7th ed. Belmont, CA: Brooks/Cole.

Washburn, A. R., & Kress, M. (2009). *Combat modeling*. New York: Springer.

Welch, B. L. (1951). On the comparison of several mean values: An alternative approach. *Biometrika, 38*(3/4), 330–336.

Wikle, C. K. (2003). *Hierarchical Models in Environmental Science.* Netherlands: International Statistical Institute.

World Air Forces, Turkey - Air Force. Retrieved from IHS Jane's: https://janes.ihs.com/WorldAirForces/Display/1319137#F-16 Peace Onyx III upgrades.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California